УДК 621.391

АНАЛІЗ ХАРАКТЕРИСТИК БЕЗПЕКИ МЕРЕЖНОГО ОБЛАДНАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОЦІНЮВАННЯ

О.С. Єременко, А.В. Спесівцев, А.В. Марчук, Л.І. Мельнікова
Харківський національний університет радіоелектроніки

Abstract *– The article addresses the urgent task of analyzing the security characteristics of network equipment and software, and of implementing an automated system to assess them, in order to support informed decision-making during the design of secure information and communication networks. It is noted that, given the significant variety of models and manufacturers and the constant increase in the number of identified vulnerabilities, the analysis and selection of optimal technical solutions are becoming more difficult. The feasibility of using an automated approach to assess the security level of network equipment based on open-source information, particularly the CVE and NVD databases and the CVSS assessment system, is justified. A web application is proposed to comprehensively analyze the security characteristics of network devices using quantitative vulnerability indicators and equipment ranking mechanisms. The software architecture has been developed to collect, store, process, and visualize data on network equipment vulnerabilities. The system is implemented using a technology stack that includes a backend based on Django and a frontend using React, TypeScript, and Tailwind CSS, as well as an object-relational PostgreSQL database. As part of the work, an analytical module was implemented to generate ratings for network devices based on security indicators, convenient interaction between the client and server was ensured via an API, and testing of the system's main functional components was carried out. The results confirm the effectiveness of the proposed approach and the feasibility of using the developed software tool in the design of secure network infrastructure. The presented solution is scalable and can be adapted to analyze different classes of network equipment, considering changes in vulnerability databases. The practical application of the developed system improves the validity of technical decisions and reduces subjectivity in selecting network components.*

Анотація *– Стаття присвячена вирішенню актуального завдання щодо аналізу характеристик безпеки мережного обладнання та програмної реалізації автоматизованої системи їхнього оцінювання з метою підтримки прийняття обґрунтованих рішень під час проєктування захищених інфокомунікаційних мереж. Зазначено, що відповідно до значної різноманітності моделей і виробників, а також постійного зростання кількості виявлених вразливостей ускладнюється аналіз і вибір оптимальних технічних рішень. Обґрунтовано доцільність застосування автоматизованого підходу до оцінювання рівня безпеки мережного обладнання на основі відкритих джерел інформації, зокрема баз CVE, NVD та системи оцінювання CVSS. Запропоновано вебзастосунок для проведення комплексного аналізу характеристик безпеки мережних пристроїв із використанням кількісних показників вразливостей і механізмів ранжування обладнання. Розроблено архітектуру програмного засобу, що забезпечує збір, зберігання, обробку та візуалізацію даних про вразливості мережного обладнання. Реалізація системи виконана з використанням технологічного стеку, що включає серверну частину на базі Django та клієнтську частину з використанням React, TypeScript і Tailwind CSS, а також об'єктно-реляційну базу даних PostgreSQL. У межах роботи реалізовано аналітичний модуль для формування рейтингів мережних пристроїв за показниками безпеки, забезпечено зручну взаємодію між клієнтською та серверною частинами через API та проведено тестування основних функціональних компонентів системи. Отримані результати підтверджують ефективність запропонованого підходу та доцільність використання розробленого програмного інструменту в процесах проєктування захищеної мережної інфраструктури. Представлене рішення є масштабованим та може бути адаптоване до аналізу різних класів мережного обладнання з урахуванням змін у базах вразливостей. Практичне застосування розробленої системи сприяє підвищенню обґрунтованості технічних рішень та зниженню рівня суб'єктивності під час вибору мережних компонентів.*

Вступ

Проєктування захищеної мережної інфраструктури потребує обґрунтованого вибору компонентів з урахуванням їхніх характеристик безпеки [1–3]. Проте через велику кількість доступних моделей маршрутизаторів, комутаторів, точок доступу та інших мережних пристроїв ручний аналіз відповідних параметрів є трудомістким, не-

точним і суб'єктивним. У зв'язку з цим виникає потреба в автоматизованих програмних засобах, що дозволяють здійснювати комплексний аналіз характеристик безпеки мережного обладнання на основі відкритих джерел, як-от база вразливостей Common Vulnerabilities and Exposures (CVE), система оцінювання Common Vulnerability Scoring System (CVSS) і репозиторій National Vulnerability Database (NVD) [4–8].

Слід зазначити, що побудова захищеної мережної інфраструктури починається не з безпосереднього вибору апаратних або програмних засобів, а з ґрунтовного визначення її призначення, цілей функціонування та принципів організації. На початковому етапі проєктування необхідно сформувати цілісне уявлення про майбутню мережу, зокрема визначити логіку взаємодії її компонентів, типи фізичних і логічних з'єднань, принципи маршрутизації трафіку та використовувані протоколи обміну даними. Формалізація зазначених аспектів створює основу для формування вимог до архітектури мережі, її масштабованості, керованості та рівня безпеки.

Лише за умови чіткого розуміння функціональних потреб, експлуатаційних обмежень і потенційних загроз доцільно переходити до вибору конкретних технічних рішень та аналізу характеристик безпеки мережного обладнання. Такий підхід дозволяє не лише підвищити ефективність проєктування, а й забезпечити системність в процесі впровадження засобів захисту та подальшій автоматизації оцінювання безпеки.

Отже, проєктування інфокомунікаційної мережі (ІКМ) базується на сукупності взаємопов'язаних складових, що формують логічну та функціональну основу мережної інфраструктури [1–3], зокрема:

- архітектурі мережі, що визначає фізичну й логічну структуру ІКМ та способи взаємодії її компонентів;
- мережних засобах і протоколах, які охоплюють апаратні компоненти мережі та правила обміну даними між вузлами;
- адресації та маршрутизації, що забезпечують унікальну ідентифікацію вузлів та ефективне передавання трафіку;
- продуктивності та масштабованості, які характеризують здатність мережі забезпечувати необхідні показники швидкодії та адаптуватися до змін навантаження;
- управлінні та безпеці, що включають процеси моніторингу, адміністрування, аналізу подій і впровадження засобів захисту мережі.

Водночас мережна безпека є ключовим чинником під час проєктування та експлуатації ІКМ, оскільки вона спрямована на забезпечення цілісності, доступності та конфіденційності каналів і вузлів мережі, а також даних, що передаються. Крім того, належний рівень мережної безпеки підвищує стійкість інфраструктури до кібератак, забезпечує безперервність роботи мережних сервісів і створює передумови для надійного та масштабованого функціонування мережі в умовах зростання навантажень і складності архітектури.

Сучасний ринок мережного обладнання налічує сотні моделей комутаторів, маршрутизаторів і точок доступу, що ускладнює вибір найбільш безпечного рішення

для побудови надійної інфраструктури. Враховуючи постійне зростання кількості вразливостей CVE та динаміку появи нових загроз, ручний аналіз технічних характеристик і звітів безпеки стає неефективним. У зв'язку з цим виникає необхідність автоматизованого підходу до оцінки безпеки мережних пристроїв, що ґрунтується на кількісному аналізі показників безпеки.

Отже, стаття присвячена актуальній прикладній задачі, пов'язаній з розробкою та тестуванням програмного інструменту для аналізу характеристик безпеки мережного обладнання з метою підтримки прийняття рішень під час проєктування захищеної мережної інфраструктури. Представлений підхід, закладений в розроблений застосунок, дозволяє формувати рейтинг пристроїв, що забезпечує прийняття обґрунтованих рішень при проєктуванні захищених ІКМ. Було запропоновано архітектуру програмного засобу щодо проведення комплексного аналізу характеристик безпеки мережного обладнання, виконано реалізацію та доведено ефективність вебзастосунку.

I. Загальний огляд програмного засобу

Для реалізації автоматизованої системи аналізу мережного обладнання за критеріями безпеки розроблюється вебзастосунок, який надає користувачам можливість переглядати, фільтрувати та оцінювати пристрої на основі даних CVE та CVSS з бази NVD [4–8]. Система дозволятиме формувати рейтинг пристроїв, обирати потрібні критерії та візуалізувати результати у зручному інтерфейсі. Для цього використано сучасний програмний стек, який охоплює зберігання даних, обчислювальну логіку, API-доступ і фронтенд-візуалізацію. Нижче наведено основні компоненти системи, їх призначення (табл. 1) та характеристика [9–15].

Таблиця 1. Технологічний стек вебзастосунку формування рейтингу пристроїв з урахуванням факторів безпеки

Рівень	Технології / Інструменти	Призначення
База даних	PostgreSQL	Зберігання пристроїв, CVE, балів CVSS, розрахунків рейтингу
Бекенд	Представлення Django та JsonResponse	API для доступу до даних, логіка обробки CVE/рейтингів
Фронтенд	React, TypeScript та Tailwind CSS	Клієнтський інтерфейс, таблиці, фільтри, графіки
Логіка обробки	Засоби Django ORM та Python	Обчислення середніх CVSS-балів, ризик-метрик

Система управління базами даних

У межах даної роботи PostgreSQL було обрано як основну систему з огляду на її функціональні переваги та сумісність з обраним технологічним стеком. Однією з ключових причин вибору стала об'єктно-реляційна модель, яка добре підходить для структурування даних щодо мережних пристроїв, уразливостей та оцінок CVSS [10]. Інтег-

рація з фреймворком Django є ще однією суттєвою перевагою. PostgreSQL добре підтримується системою Object-Relational Mapping (ORM), що входить до складу Django, завдяки чому взаємодія з базою даних (БД) реалізується через зручні Python-моделі [9, 11]. Крім того, PostgreSQL відзначається високим рівнем стабільності та розширюваності: її архітектура дозволяє додавати нові таблиці, типи зв'язків і процедури, що зберігаються, для реалізації складної логіки в процесі подальшого розвитку проєкту.

Отже, в PostgreSQL зберігається інформація про мережні пристрої, зокрема модель, тип і виробник, а також перелік відомих CVE. Таким чином, ця система управління базами даних (СУБД) відіграє центральну роль у забезпеченні зберігання, обробки та доступу до структурованих даних, необхідних для ефективної роботи рейтингової системи безпеки мережних пристроїв.

Серверна частина

Серверна логіка розробленого вебзастосунку реалізована виключно засобами Django (табл. 2) [9, 11, 12]. Передача даних між клієнтською та серверною частинами (зокрема, реалізованою за допомогою React) здійснюється через HyperText Transfer Protocol (HTTP)-запити, на які сервер відповідає у форматі JavaScript Object Notation (JSON). Для цього використовуються стандартні представлення Django (views), які обробляють вхідні параметри, виконують необхідну бізнес-логіку та формують відповідь за допомогою об'єкта JsonResponse.

Однією з переваг Django є тісна інтеграція з БД PostgreSQL: структура бази моделюється у вигляді Python-класів (моделей), які автоматично синхронізуються з таблицями БД. Такий підхід дозволяє легко оперувати даними, будувати фільтрації та агрегації на рівні запитів до бази, не використовуючи сторонні інструменти.

У проєкті реалізовано низку REST-подібних API кінцевих точок (endpoint), які забезпечують доступ до інформації про пристрої, вразливості, рейтинги, а також дозволяють виконувати фільтрацію, сортування та аналітичну обробку даних. Серверна частина підтримує обчислення середніх показників BaseScore, формування рейтингів та інші операції, необхідні для оцінки безпеки.

Таблиця 2. Реалізація функцій Django та REST Framework у проєкті

Функція	Реалізація
Зберігання пристроїв та CVE	Django-моделі та PostgreSQL ORM
Розрахунок рейтингів	Засоби Django ORM та Python
API для фронтенду	Представлення Django та JsonResponse

Клієнтська частина

Клієнтську частину програмного засобу реалізовано з використанням сучасного фронтенд-стеку, що включає React, TypeScript та Tailwind CSS (табл. 3) [13–15]. Обрані технології забезпечують створення швидкого, адаптивного та масштабованого вебінтерфейсу з підтримкою динамічного контенту.

1) React застосовано для побудови компонентної архітектури інтерфейсу та реалізації односторінкового застосунку (Single Page Application, SPA), що забезпечує ефективне оновлення вмісту без перезавантаження сторінок. У межах клієнтської частини реалізовано відображення й обробку даних про мережні пристрої, включно з сортуванням і фільтрацією за показниками CVSS, а також візуалізацію рейтингових показників і результатів аналізу безпеки.

2) TypeScript використано для забезпечення статичної типізації та перевірки коректності даних, отриманих через API, що підвищує надійність коду та спрощує його подальший супровід.

3) Tailwind CSS застосовано для реалізації адаптивного інтерфейсу користувача та уніфікованого дизайну без надмірного обсягу стилів, що сприяє швидкій розробці та підтримці інтерфейсу.

Обмін даними між клієнтською та серверною частинами здійснюється за допомогою HTTP-запитів до API, що забезпечує динамічне оновлення інформації та узгоджену взаємодію між компонентами системи.

Таблиця 3. Переваги обраного фронтенд-стеку

Компонент	Переваги
React	Швидкий, компонентний підхід, динамічні таблиці й графіки
TypeScript	Безпечна робота з даними, статична типізація, автодоповнення
Tailwind CSS	Мінімум коду для стилів, адаптивність, чистий дизайн

Аналітичний модуль

Для реалізації логіки розрахунку рейтингу мережних пристроїв за критеріями безпеки у серверній частині застосовано стандартні інструменти фреймворку Django та мови програмування Python [9]. Аналітична обробка здійснюється за допомогою механізмів, що надає ORM Django, зокрема агрегатних функцій (Avg, Count) та фільтрації через об’єкти запитів (QuerySet).

Основу модуля становить підхід до обробки даних безпосередньо на рівні бази даних PostgreSQL. Замість завантаження великих обсягів інформації в оперативну пам’ять та обробки її зовнішніми бібліотеками, обчислення середніх значень, підрахунки кількості вразливостей та сортування пристроїв реалізовано через SQL-запити, сформовані за допомогою Django ORM. Такий підхід дозволяє ефективно масштабувати рішення та зменшує навантаження на сервер.

Серед функцій аналітичного модуля – визначення кількості вразливостей у розрізі виробників і типів пристроїв, формування десяти найбільш критичних пристроїв за середнім показником CVSS BaseScore, а також виведення найменш уразливих пристроїв. Розрахунки проводяться з урахуванням додаткових метрик, як-от ExploitabilityScore й ImpactScore, які також агрегуються у відповідних запитах.

Сортування, фільтрація та формування зведених даних здійснюється програмно на рівні Python із використанням базових структур (словників, списків, умовних операторів), що забезпечує прозору й контрольовану логіку обробки без залучення сторонніх залежностей (табл. 4).

Таблиця 4. Переваги реалізації аналітики засобами Django

Перевага	Пояснення
Продуктивність	Використання агрегатів ORM дозволяє обробляти великі обсяги CVE-даних без потреби в додаткових бібліотеках
Простота	Реалізація логіки засобами Python і Django без зовнішніх залежностей
Гнучкість	Підтримка складних фільтрів, сортування та логіки ранжування
Інтеграція з API	Зручне формування відповідей у форматі JSON для клієнтської частини

II. Опис підходу до оцінювання безпеки мережних пристроїв

Для обґрунтованого вибору мережного обладнання в контексті побудови захищеної інфокомунікаційної інфраструктури необхідним є попередній аналіз факторів безпеки. В рамках даної розробки за основу взято базу даних вразливостей NVD, що містить відкриті записи CVE з відповідними оцінками за системою CVSS версії 3.x (зокрема 3.1) для пристроїв Cisco, Huawei та Juniper [4–8].

Основними параметрами, які використовуються для оцінки пристроїв, є:

- назва пристрою (Device);
- виробник (Vendor);
- тип пристрою (Type);
- числові показники безпеки, отримані з CVSS (BaseScore, ExploitabilityScore та ImpactScore).

Ці параметри дозволяють проводити автоматизований кількісний аналіз, що значно прискорює процес формування рейтингів безпеки великої кількості пристроїв:

- BaseScore залежить від впливу та експлуатованості уразливості;
- ExploitabilityScore визначає, наскільки легко реалізувати атаку;
- ImpactScore оцінює потенційні наслідки компрометації пристрою.

Окрему роль у процесі аналізу відіграє CVSS-вектор (Vector String), який містить деталізовану технічну інформацію про кожну уразливість: тип вектора атаки (локальний, мережний), необхідний рівень привілеїв, взаємодію з користувачем, вплив на конфіденційність, цілісність і доступність тощо.

Унікальність запропонованого підходу полягає в тому, що користувач самостійно може обирати критерії оцінювання обладнання, застосовуючи фільтри до компонентів CVSS. Це надає гнучкість у прийнятті рішень залежно від конкретних вимог до безпеки.

На основі відфільтрованих і оцінених даних система автоматично формує рейтинг мережних пристроїв, а також узагальнює результати у вигляді рейтингу виробників і типів обладнання (наприклад, маршрутизаторів, комутаторів, точок доступу). Це дозволяє швидко виявити найбільш ризиковані або, навпаки, рекомендовані до використання рішення в інфраструктурі.

III. Розробка програмного засобу для формування рейтингу пристроїв з урахуванням параметрів безпеки

Вебінтерфейс

Клієнтську частину програмного засобу реалізовано у вигляді односторінкового застосунку (SPA, рис. 1) [9]. Такий підхід забезпечує зручний і зрозумілий інтерфейс для доступу до функцій системи аналізу характеристик безпеки мережного обладнання.

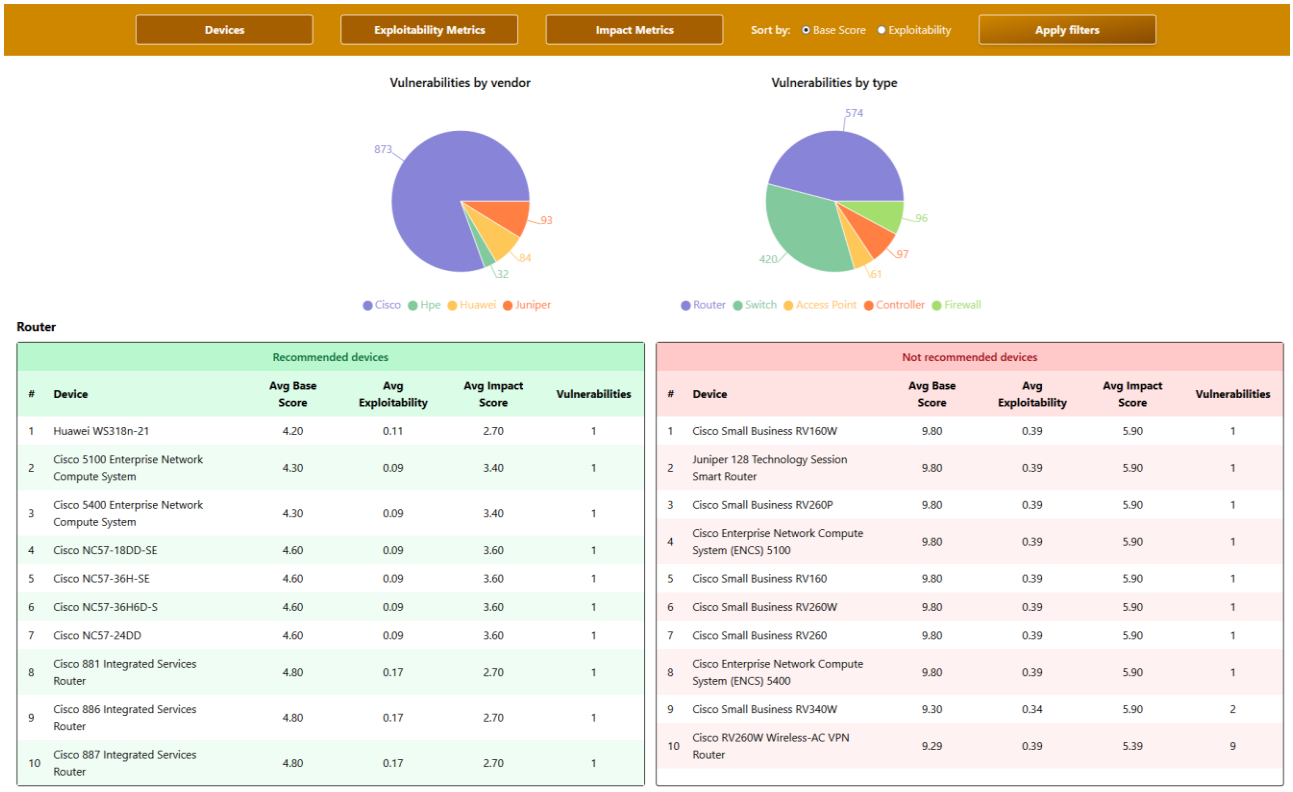

Рис. 1. Загальний вигляд інтерфейсу

Інтерфейс надає користувачу можливість застосовувати множинні фільтри для звуження вибірки пристроїв (рис. 2). Фільтрація виконується за параметрами CVSS-вектора (наприклад, тип вектора атаки, необхідні привілеї, взаємодія з користувачем), а також за типом і виробником пристроїв.

Рис. 2. Панель фільтрації та сортування

Головна частина інтерфейсу представлена у вигляді табличного компонента на основі типу пристроїв (рис. 3 – 7), що містить основну інформацію про модель, виробника, тип обладнання, середні показники безпеки та кількість виявлених вразливостей. Користувач має можливість переглядати рейтинг пристроїв за обраними критеріями.

Router

Recommended devices

#	Device	Avg Base Score	Avg Exploitability	Avg Impact Score	Vulnerabilities
1	Huawei WS318n-21	4.20	0.11	2.70	1
2	Cisco 5100 Enterprise Network Compute System	4.30	0.09	3.40	1
3	Cisco 5400 Enterprise Network Compute System	4.30	0.09	3.40	1
4	Cisco NC57-18DD-SE	4.60	0.09	3.60	1
5	Cisco NC57-36H-SE	4.60	0.09	3.60	1
6	Cisco NC57-36H6D-S	4.60	0.09	3.60	1
7	Cisco NC57-24DD	4.60	0.09	3.60	1
8	Cisco 881 Integrated Services Router	4.80	0.17	2.70	1
9	Cisco 886 Integrated Services Router	4.80	0.17	2.70	1
10	Cisco 887 Integrated Services Router	4.80	0.17	2.70	1

Not recommended devices

#	Device	Avg Base Score	Avg Exploitability	Avg Impact Score	Vulnerabilities
1	Cisco Small Business RV160W	9.80	0.39	5.90	1
2	Juniper 128 Technology Session Smart Router	9.80	0.39	5.90	1
3	Cisco Small Business RV260P	9.80	0.39	5.90	1
4	Cisco Enterprise Network Compute System (ENCS) 5100	9.80	0.39	5.90	1
5	Cisco Small Business RV160	9.80	0.39	5.90	1
6	Cisco Small Business RV260W	9.80	0.39	5.90	1
7	Cisco Small Business RV260	9.80	0.39	5.90	1
8	Cisco Enterprise Network Compute System (ENCS) 5400	9.80	0.39	5.90	1
9	Cisco Small Business RV340W	9.30	0.34	5.90	2
10	Cisco RV260W Wireless-AC VPN Router	9.29	0.39	5.39	9

Рис. 3. Приклад табличного компонента для типу Router

Вебінтерфейс включає графічні компоненти, реалізовані за допомогою React-бібліотек для побудови діаграм, що дозволяють візуально порівнювати кількість унікальних вразливостей за виробниками та типами пристроїв (рис. 4).

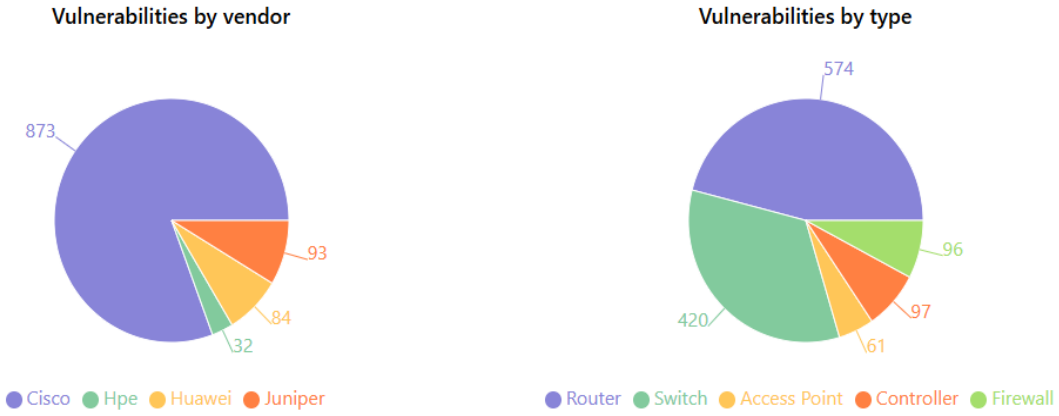

Рис. 4. Візуалізація кількості унікальних вразли

Створення API-інтерфейсу та елементів фільтрації

У межах розробленого програмного засобу реалізовано API-інтерфейс, що забезпечує взаємодію між клієнтською частиною та серверною логікою системи. Для цього використано базові можливості фреймворку Django без залучення сторонніх REST-бібліотек. Обробка запитів реалізована через функції-представлення (views), які приймають HTTP-запити, виконують відповідну бізнес-логіку та повертають результати у форматі JSON.

Кожна API кінцева точка є окремою функцією, що обробляє параметри запиту, виконує фільтрацію та агрегацію даних з використанням можливостей Django ORM, після чого формує відповідь у вигляді об'єкта JsonResponse. Такий підхід дозволяє реалізувати REST-подібну архітектуру із підтримкою стандартних HTTP-методів (переважно GET) та передачі даних у форматі JSON.

Функціональність API охоплює:

– отримання кількості вразливостей за типами пристроїв та виробниками;
– формування зведеної інформації про пристрої з найвищими та найнижчими середніми показниками ризику;
– обробку запитів із фільтрами, які задаються користувачем у клієнтському інтерфейсі.

Всі запити до API є синхронними, і обробка виконується в реальному часі, що забезпечує оперативне оновлення аналітичних даних у вебінтерфейсі. Завдяки використанню вбудованих засобів Django досягається простота реалізації, легкість підтримки коду та можливість розширення функціоналу у майбутньому без необхідності суттєвої переробки архітектури.

Основні API кінцеві точки:

– `count_vulnerabilities_by_vendor/` повертає кількость унікальних вразливостей, на пристроях за виробниками;
– `count_vulnerabilities_by_type/` повертає кількость унікальних вразливостей, на пристроях за типом;
– `top_10_devices_by_score/` повертає список з десяти найбільш та найменш вразливих пристроїв.

У рамках реалізації системи обробки даних про вразливості пристроїв було запроваджено гнучкий механізм фільтрації та сортування, який забезпечує динамічне формування запитів до бази даних відповідно до обраних користувачем критеріїв. Фільтрація реалізована на рівні серверної логіки з використанням конструкцій умовного об'єднання (об'єктів Q) у середовищі Django ORM.

Під час обробки запиту користувача застосовуються фільтри для двох основних сутностей: вразливості (Vulnerability) та мережні пристрої (Device). Для кожного з цих параметрів можуть бути задані множинні значення (рис. 5). Якщо жодне значення не вказано, фільтр до відповідного поля не застосовується. Для сортування в запиті передається параметр `sort_by`, який визначає критерій сортування пристроїв (рис. 6) у таблицях за типом: `base_score` або `exploitability`. Для застосування фільтрів та сортування потрібно натиснути кнопку Apply filters (рис. 7), яке передає обрані параметри до серверної частини (бекенду).

Модель даних і структура бази

Для реалізації програмного засобу з аналізу характеристик безпеки мережного обладнання розроблено реляційну модель даних, яка відображає взаємозв'язки між пристроями, їх виробниками, типами та відомими вразливостями. Структура бази даних є критичною складовою архітектури системи, оскільки вона забезпечує збереження, доступність і цілісність даних, необхідних для формування рейтингів безпеки.

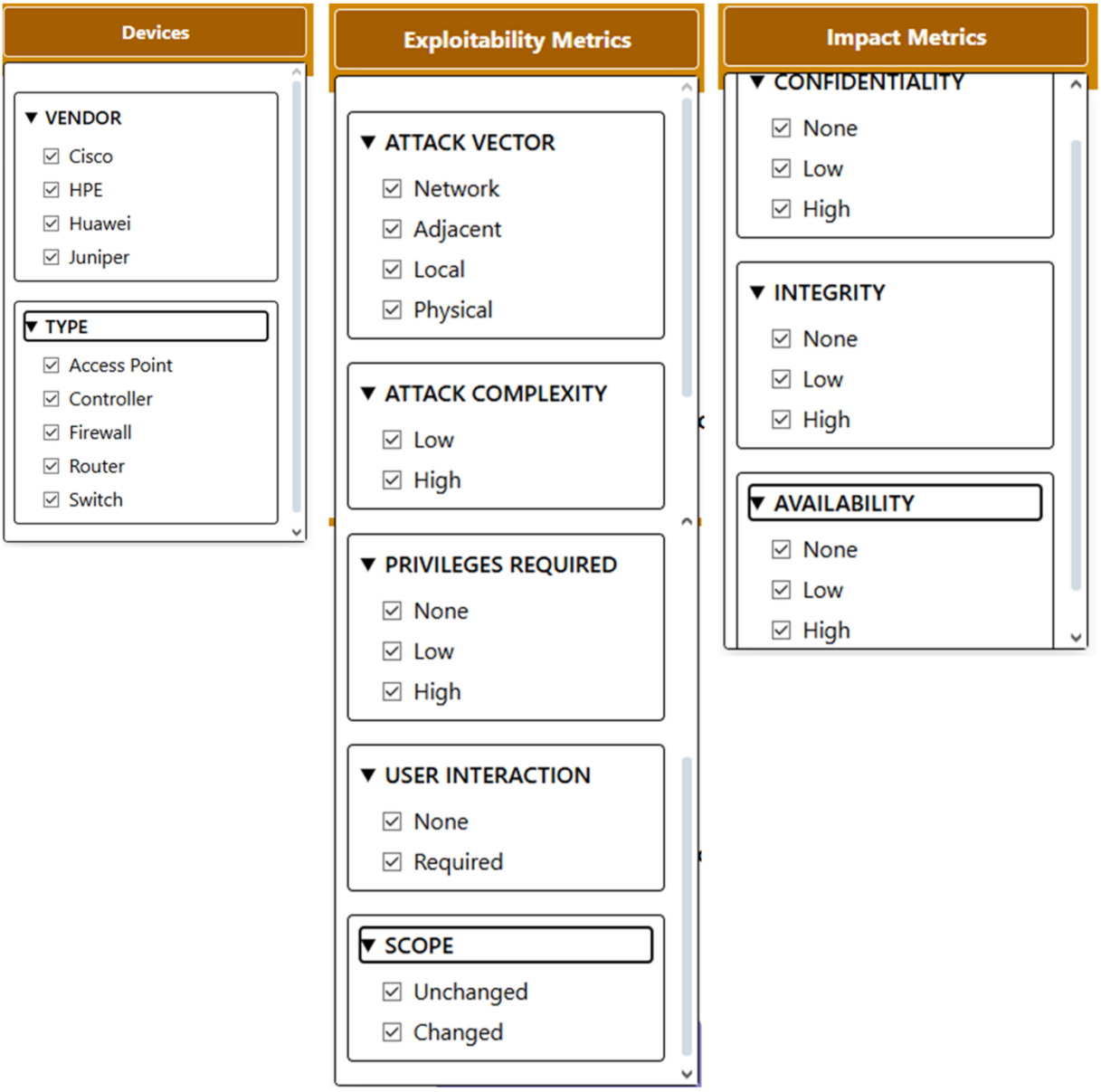

Рис. 5. Параметри фільтрації категорії пристроїв

Рис. 6. Критерії сортування

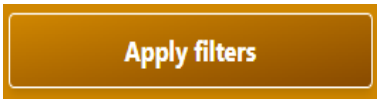

Рис. 7. Кнопка для застосування фільтрів

Обрана СУБД PostgreSQL підтримує складні зв'язки, індексацію, агрегацію та запити з високим рівнем продуктивності. За допомогою вбудованої ORM фреймворку Django реалізовано логічне відображення таблиць бази на Python-класи, що полегшує доступ до даних у серверній частині (рис. 8).

Таблиця `vulnerabilities_device` є ключовою та містить інформацію про мережне обладнання (назва, виробник та тип). Кожен запис має унікальний ідентифікатор і пов'язаний з таблицями:

- `vulnerabilities_vendor`;
- `vulnerabilities_type`.

Таблиця `vulnerabilities_vulnerability` є ключовою та містить інформацію про вразливості (назва CVE, BaseScore, ExploitabilityScore, ImpactScore, Attack Vector, Attack Complexity, Privileges Required, User Interaction, Scope, Confidentiality Impact, Integrity Impact, Availability Impact). Кожен запис має унікальний ідентифікатор і пов'язаний з таблицями:

- `vulnerabilities_attack_complexity`;
- `vulnerabilities_attack_vector`;
- `vulnerabilities_availability`;
- `vulnerabilities_confidentiality`;
- `vulnerabilities_integrity`;
- `vulnerabilities_privileges_required`;
- `vulnerabilities_scope`;
- `vulnerabilities_user_interaction`.

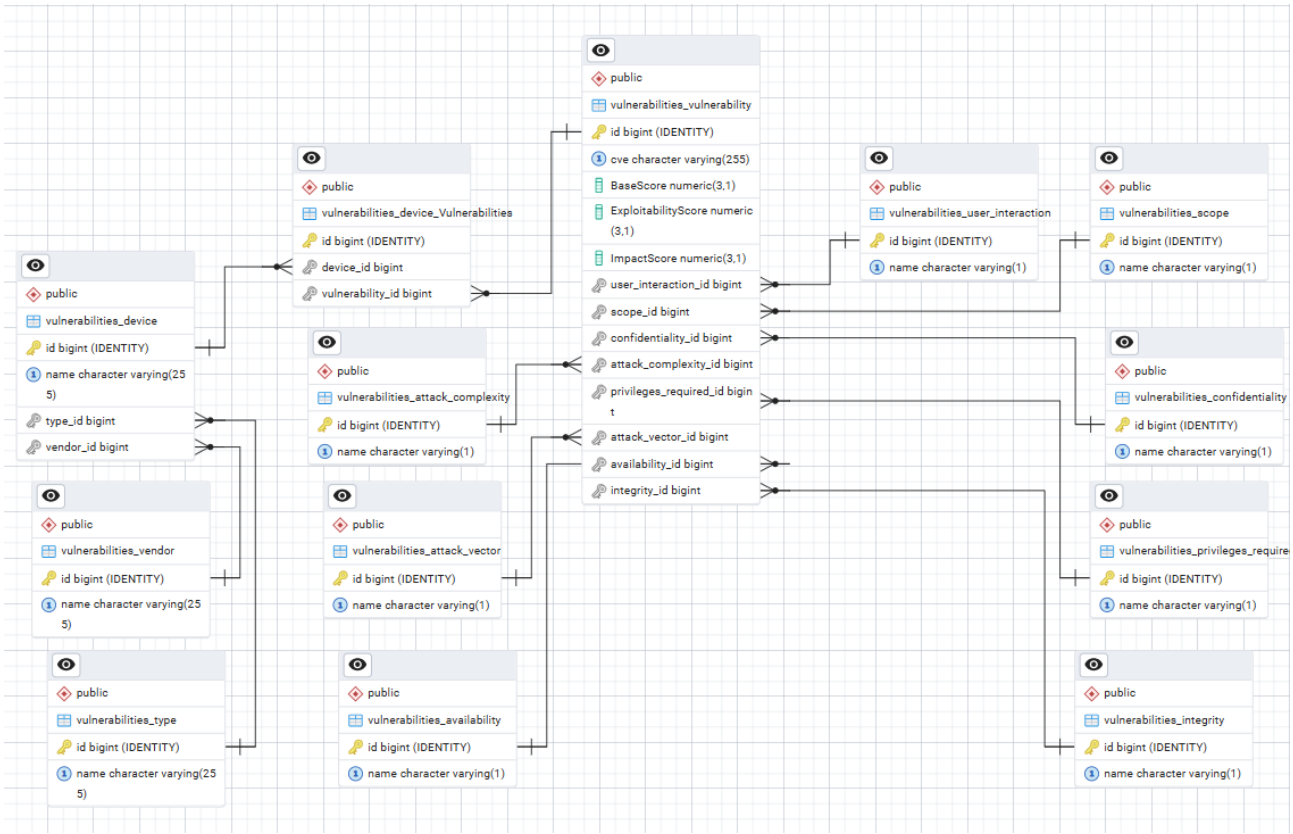

Рис. 8. Діаграма «сутність – зв'язок» (Entity-Relationship Diagram, ERD)

Таблиця `vulnerabilities_device_Vulnerabilities` створена для реалізації зв'язку «багато до багатьох» між таблицями `vulnerabilities_device` та `vulnerabilities_vulnerability`. Один пристрій може мати декілька вразливостей, і одна вразливість може стосуватися кількох пристроїв.

Ця модель (рис. 8) забезпечує масштабованість системи, дозволяє легко додавати нові типи пристроїв, розширювати характеристики вразливостей або інтегрувати дані з нових джерел. Структура бази також оптимізована під виконання аналітичних запитів – наприклад, виведення найбільш критичних вразливостей або формування рейтингів з урахуванням обраних критеріїв.

Реалізація аналітичного модуля

Аналітичний модуль розробленого програмного засобу реалізовано на основі засобів Django ORM, що дозволяє ефективно обробляти запити до бази даних без використання сторонніх бібліотек для аналізу даних. Основною метою модуля є формування аналітичних звітів щодо вразливостей пристроїв, згрупованих за різними критеріями.

Для отримання узагальненої інформації про кількість вразливостей реалізовано окремі функції, які здійснюють агрегацію даних за відповідними фільтрами. Зокрема, функція count_vulnerabilities_by_vendor виконує підрахунок кількості унікальних вразливостей, що стосуються пристроїв кожного з виробників. Аналогічно, функція count_vulnerabilities_by_type виконує підрахунок за типами пристроїв. У кожному випадку реалізовано фільтрацію та сортування за допомогою функції data_by_filters, яка враховує передані параметри запиту.

Окремий функціональний блок представлений у вигляді методу top_10_devices_by_score, що формує зведені дані про пристрої з найвищими та найнижчими середніми Base Score або Exploitability. Обробка здійснюється у розрізі типів пристроїв, для кожного з яких формується два окремі списки: топ-10 найбільш критичних та нижні 10 найменш вразливих пристроїв. Для кожного пристрою додатково обчислюються середні показники впливу (Impact Score), а також загальна кількість пов'язаних вразливостей.

Усі обчислення виконуються на рівні запитів до бази даних із використанням агрегатних функцій Avg та Count, що забезпечує оптимальну продуктивність без потреби в завантаженні великих обсягів даних у оперативну пам'ять. Результати аналітики повертаються у форматі JSON, що дозволяє легко інтегрувати їх у вебінтерфейс користувача.

Завдяки використанню Django ORM забезпечено ефективність та масштабованість обробки даних, а також зручність у модифікації логіки агрегації без порушення цілісності програми.

Основні функції аналітичного модуля:

1) Обчислення середніх показників безпеки. Для кожного пристрою система розраховує середнє значення наступних CVSS-показників: BaseScore, Exploitability, ImpactScore.

2) Аналітичний модуль дозволяє виконувати фільтрацію вразливостей за типом, виробником та окремими компонентами CVSS-вектора, такими як: Attack Vector (AV), Privileges Required (PR), User Interaction (UI), Scope, Confidentiality, Integrity, Availability тощо. Таке фільтрування дає змогу адаптувати аналітику під специфічні вимоги безпеки.

3) Після виконання обчислень, результати формуються у форматі структурованих об'єктів JSON, які передаються через REST API на клієнтську частину для подальшої візуалізації. За потреби результати можуть бути кешовані або агреговані для підвищення продуктивності.

Тестування та налагодження функціоналу

Забезпечення коректної, стабільної та передбачуваної роботи програмного засобу є критично важливим етапом у процесі його розробки. Для досягнення зазначених цілей у межах реалізації вебсистеми для аналізу характеристик безпеки мережного обладнання було проведено комплексне тестування та налагодження функціоналу на рівнях даних, логіки обробки, API та інтерфейсу користувача.

Тестування окремих компонентів серверної частини (моделей, представлень API, логіки аналітики) реалізовано із використанням вбудованих інструментів фреймворку Django (unittest, pytest-django). Основні сценарії, що тестуються:

- коректність створення об'єктів моделей (пристрої, уразливості, рейтинги);
- обробка edge-кейсів (нульові значення, дублікати CVE);
- правильність обчислень;
- відповідність структури API-відповідей очікуваним схемам (формат JSON).

Усі критично важливі функції аналітичного модуля проходять окремі тести на точність математичних операцій, включно з агрегацією, фільтрацією, нормалізацією та ранжуванням.

Для перевірки взаємодії між сервером і клієнтом реалізовано інтеграційне тестування API. Воно охоплює такі аспекти:

- доступність основних ендпоінтів;
- коректна обробка запитів з параметрами;
- тестування пагінації, сортування, фільтрації та кодування відповіді.

Клієнтська частина проходить ручне та автоматизоване тестування з метою перевірки візуальної коректності, інтерактивності та відповідності поведінки інтерфейсу очікуваній логіці. Серед перевірених сценаріїв:

- правильність завантаження та відображення таблиці пристроїв;
- фільтрація пристроїв за параметрами типу, виробника та CVSS;
- відображення динамічних графіків та рейтингів;
- реагування інтерфейсу на відсутність даних чи помилки запиту.

Висновки

В роботі вирішено задачу щодо розробки програмного інструменту для аналізу характеристик безпеки мережного обладнання з метою підтримки прийняття рішень при проєктуванні захищеної мережної інфраструктури.

З цією метою було проаналізовано наявні підходи до оцінювання характеристик безпеки мережного обладнання, сучасні методи аналізу вразливостей, особливості представлення інформації в базах CVE, CVSS і NVD, а також інструменти, які можуть бути використані для побудови подібних систем. Показано, що на теперішній час актуальною є потреба в автоматизованих засобах, які дозволяють швидко, достовірно та на основі відкритих даних оцінювати ризики, пов'язані з використанням мережного обладнання, та забезпечувати прийняття обґрунтованих рішень під час проєктування. З огляду на актуальність проблеми зростання кількості вразливостей та необхідність оперативного реагування на кіберзагрози, запропоновано підхід до ранжування мережних пристроїв за критеріями безпеки на основі даних з баз CVE, CVSS та NVD.

Розроблено програмний засіб у вигляді вебзастосунку, який реалізує повний цикл роботи з інформацією про вразливості мережного обладнання: від збору даних і зберігання у базі PostgreSQL до обробки, аналітики та візуалізації результатів через зручний інтерфейс. Система реалізована із застосуванням сучасного технологічного стеку, включно з Django, React, TypeScript і Tailwind CSS. Запропоновано структуру бази даних, реалізовано API для взаємодії між клієнтською та серверною частинами, а також аналітичний модуль для формування рейтингів. Проведено тестування основних функціональних компонентів розробленої системи, яке підтвердило її працездатність, коректність обробки даних і придатність до використання в задачах проєктування захищеної інфокомунікаційної інфраструктури.

Результати роботи доцільно використовувати для подальшого вдосконалення процесів оцінювання ризиків інформаційної безпеки, інтеграції з іншими системами кіберзахисту, а також у практиці прийняття технічних рішень щодо вибору обладнання в умовах підвищених вимог до захищеності мереж.

Список літератури

1. *Лемешко, О. В., Єременко, О. С., Невзорова, О. С.* (2020), Потокові моделі та методи маршрутизації в інфокомунікаційних мережах: відмовостійкість, безпека, масштабованість, Харків: ХНУРЕ, 308 с. **DOI**: https://doi.org/10.30837/978-966-659-282-1.

2. *Лемешко, О.В., Єременко, О.С., Євдокименко, М.О., Шаповалова, А.С., Слейман, Б.* (2022), Моделювання та оптимізація процесів безпечної та відмовостійкої маршрутизації в телекомунікаційних мережах, Харків: ХНУРЕ, 198 с. **DOI**: https://doi.org/10.30837/978-966-659-378-1.

3. Network Design Guide – What are the elements of Network Design. **URL**: https://macronetservices.com/network-design-guide-what-are-the-elements-of-network-design/ .

4. What Is the National Vulnerability Database (NVD)? **URL**: https://www.fortinet.com/lat/resources/cyberglossary/national-vulnerability-database-nvd.

5. CVE and CVSS: Vulnerability Assessment & Severity Ratings. **URL**: https://www.opus.security/blog/cve-cvss.

6. Common Vulnerability Scoring System Calculator. **URL**: https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator.

7. Common Vulnerability Scoring System v3.1: Specification Document. **URL**: https://www.first.org/cvss/v3-1/specification-document.

8. *Balsam, A., Nowak, M., Walkowski, M., Oko, J., Sujecki, S.* (2024), "Comprehensive comparison between versions CVSS v2.0, CVSS v3.x and CVSS v4.0 as vulnerability severity measures", Proceedings of the 2024 24th International Conference on Transparent Optical Networks (ICTON), Bari, Italy, P. 1–4, **DOI**: https://doi.org/10.1109/ICTON62926.2024.10647452.

9. *Fenollosa A.* (2022), Building SPAs with Django and HTML Over the Wire: Learn to build real-time single page applications with Python, Packt Publishing, 264 p.

10. *Ferrari, L., Pirozzi, E.* (2023), Learn PostgreSQL: Use, manage, and build secure and scalable databases with PostgreSQL 16, Packt Publishing Ltd, 744 p.

11. *Vincent, W.S.* (2025), Django for APIs, 5th Edition: Build Web APIs with Python and Django, Still River Press, 216 p.

12. *Kohl, R.* (2025), Practical Django REST Framework: Build Secure, Scalable, and Production-Ready APIs for Web and Mobile Applications with Real-word problems, Kindle Edition, 289 p.

13. *Schwarzmüller, M.* (2025), React Key Concepts: An in-depth guide to React's core features. Packt Publishing Ltd, 544 p.

14. *Despoudis, T.* (2025), TypeScript 5 Design Patterns and Best Practices: Build clean and scalable apps with proven patterns and expert insights, Packt Publishing Ltd, 424 p.

15. *Bhat, K.* (2023), Ultimate Tailwind CSS Handbook: Build sleek and modern websites with immersive UIs using Tailwind CSS, Orange Education Pvt Limited, 390 p.