тувацьких атрибутів, дають змогу здійснювати перевірку прав доступу без необхідності зберігати сесію на сервері. Застосування ролей та політик доступу забезпечує гнучке управління дозволами в межах системи, а інтеграція токена в Cookies з налаштуваннями безпеки мінімізує ризик атак типу XSS та CSRF [3].

Отже, проведене проєктування та впровадження дали змогу створити модель, що забезпечує як безпечне зберігання облікових даних, так і контроль доступу на основі ролей без постійного зберігання сесійної інформації. Така архітектура відповідає актуальним вимогам до захисту персональних даних, є масштабованою, ефективною для розподілених систем і може бути адаптована для різних типів користувацьких інтерфейсів. Запропонований підхід доводить доцільність поєднання гешування та токен-орієнтованої авторизації як основи для побудови надійної та гнучкої системи безпеки у корпоративному програмному забезпеченні.

Список використаних джерел

1. A TOTP-based secure data storage system in the cloud environment using the JWT token approach / A. Shahnawaz, A. Mohd, A. Javed, M. Shabana. *International Journal of Systems Assurance Engineering and Management.* 2025. Vol. 16. P. 1565–1578. DOI: 10.1007/s13198-025-02775-8.

2. Enhancing the Security of JSON Web Token Using Signal Protocol and Ratchet System / P. Singh, G. Choudhary, S. K. Shandilya, V. Sihag. *Proceedings of Emerging Trends and Technologies on Intelligent Systems. Advances in Intelligent Systems and Computing.* Vol. 1414. Springer, Singapore. DOI: 10.1007/978-981-19-4182-5_31.

3. Pyroh M., Tereshchuk G., Toroshanko O. Authentication Principles as Security aspects of Web Development. *Measuring And Computing Devices In Technological Processes*. 2025. №. 1. P. 294–301. DOI: 10.31891/2219-9365-2025-81-36.

УДК 004.056.5

Первачук Р. Ю., *здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,*
Антонов Ю. С., *канд. фіз.-мат. наук, доцент, доцент кафедри інформаційних технологій*

РОЗРОБКА УТИЛІТ ДЛЯ АГРЕГАЦІЇ ІНТЕРНЕТ-РЕСУРСІВ, ЩО ПІДЛЯГАЮТЬ БЛОКУВАННЮ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному інформаційному просторі за допомогою інтернет-ресурсів здійснюються різноманітні кібератаки, від фішингу до ransomware або DDoS-атак [1–3]. У зв'язку з цим підтримка актуального переліку інтернет-ресурсів, що підлягають блокуванню, має здійснюватись як на основі локальних списків, так і на основі списків, доступних у мережі Інтернет. Блокування таких джерел є одним із базових засобів протидії кіберзагрозам як у державних, так і в приватних мережах. Проте наявні засоби фільтрації часто є вузькоспеціалізованими, обмеженими у форматах або залежними від хмарних сервісів, що знижує рівень автономності та адаптивності.

Метою цієї роботи є створення модульної системи утиліт, яка виконує повний цикл: від завантаження та обробки даних про шкідливі домени до формування форматів блокування, придатних для використання у реальному середовищі. Об’єкт дослідження – процеси обробки, агрегації та фільтрації інтернет-ресурсів. Предмет – алгоритми та структури даних, що реалізують ці процеси.

Для досягнення мети були сформульовані та реалізовані такі задачі:

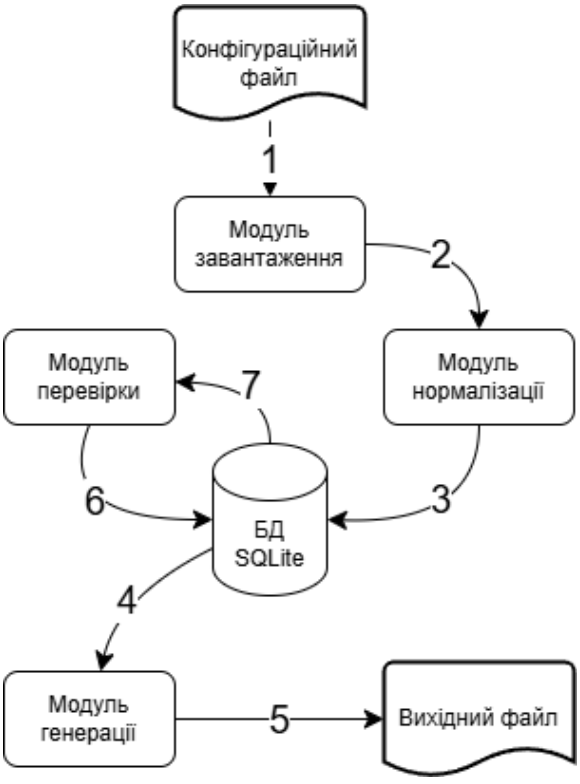
- виявлено типи загроз і механізми їх поширення, у тому числі через DNS;
- узагальнено сучасні підходи до фільтрації (локальні, хмарні, автономні);
- розроблено архітектуру системи та data-flow модель;
- спроєктовано структуру бази даних та реалізовано механізми нормалізації;
- обґрунтовано формати вихідних даних (hosts, dnsmasq, csv);
- розроблено алгоритми перевірки доступності ресурсів (ping, DNS).

Система реалізується мовою Python з використанням бібліотек: requests, PyYAML, sqlite3, dns.resolver, ping3. Зберігання даних здійснюється у SQLite, що забезпечує автономність і гнучке керування вибірками. Для генерації вихідних файлів підтримуються класичні текстові формати hosts, dnsmasq.conf, а також csv для перегляду й аналізу[4–5].

Архітектура системи має модульну будову. Усі компоненти працюють незалежно:

- модуль агрегації зчитує конфігурацію і завантажує фіди;
- модуль парсингу нормалізує записи;
- база даних виступає централізованим сховищем;
- утиліта генерації формує вихідні файли згідно з параметрами;
- модуль перевірки тестує доступність ресурсів.

Для наочності зобразимо на рисунку:



підпис!

У підсумку реалізовано прототип автономної системи фільтрації, яка дає змогу збирати та аналізувати дані з відкритих джерел, гнучко керувати категоріями загроз, а також автоматично оновлювати списки блокування для використання на комп'ютерах і маршрутизаторах.

Список використаних джерел

1. Kosinski M. What is ransomware? *International Business Machines.* URL: https://www.ibm.com/think/topics/ransomware
2. What is a Botnet? *CrowdStrike.* URL: https://www.crowdstrike.com/.../botnets/
3. Антонов Ю. С., Римар П. В., Антонова О. Г. Проблема DoS/DDoS атак навчальних ресурсів здобувачами. *Сучасний захист інформації.* 2019. № 4(40). С. 52–62. DOI: 10.31673/2409-7292.2019.045262.
4. Документація Pi-Hole. URL: https://github.com/pi-hole/docs
5. Dnsmasq. *ArchWiki.* URL: https://wiki.archlinux.org/title/Dnsmasq

УДК 004.738.1

Підруцький Д. А., *здобувач вищої освіти 4 курсу спеціальності 122 Комп'ютерні науки,*
Січко Т. В., *канд. техн. наук, доцент, доцент кафедри інформаційних технологій*

РОЗРОБКА ВЕБСАЙТА ДЛЯ ПРОДАЖУ ВІЙСЬКОВОГО СПОРЯДЖЕННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

У контексті триваючої війни особливої важливості набуває забезпечення військових якісним спорядженням. Традиційні канали закупівлі не завжди здатні оперативно задовольнити попит, тому виникає потреба у створенні спеціалізованих онлайн-ресурсів для продажу військового обладнання.

Актуальність посилюється потребою у безпечному та масштабованому середовищі для обробки великого обсягу критичних даних.

Наразі, активно досліджуються питання побудови вебплатформ на основі технологій, як от MongoDB, Express, React. Особлива увага приділяється захисту персональних даних JWT, bcrypt, структурі REST API, та інтеграції платіжних систем, як от Stripe, у системи електронної комерції.

Метою роботи є розробка повнофункціонального вебсайта військового спорядження з акцентом на забезпеченні безпеки користувачів, масштабованості та зручного користувацького інтерфейсу.

Постановка задачі:

– створити backend-сервер на Node.js з використанням Express;

– спроєктувати та реалізувати гнучку структуру бази даних на MongoDB;

– побудувати frontend-інтерфейс на основі React із використанням Tailwind CSS, Axios, React Router DOM і React Toastify;

– забезпечити безпеку автентифікації, збереження даних і реалізувати платіжну систему Stripe;