ТЗ. №4(36)'2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

Кібербезпека
та захист критичної інфраструктури

УДК 004.056:004.852

АВТОМАТИЗОВАНИЙ ПОШУК XSS-ВРАЗЛИВОСТЕЙ У ВЕБЗАСТОСУНКАХ НА ОСНОВІ МУЛЬТИАГЕНТНОГО ПІДХОДУ

В.С. Кравчук[1], Н.О. Маслова[1, 2], Я.Ю. Дорогий[1, 3, 4]
[1]Department of Applied Mathematics and Informatics, Donetsk National Technical University, Drohobych, Ukraine
[2]Department of Information Security Management, Lviv State University of Life Safety, Lviv, Ukraine
[3]Department of Information Systems and Technologies, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
[4] Department of Intelligent Software Systems, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

АНОТАЦІЯ

З розвитком інформаційних технологій та зростанням обсягів обробки чутливої інформації в Інтернеті вебзастосунки стали важливою частиною сучасних бізнес-процесів. Однак з цим збільшується і кількість кіберзагроз, що ставить перед організаціями завдання забезпечення безпеки своїх вебресурсів. Одним з основних методів захисту є тестування на проникнення, яке дозволяє виявити вразливості шляхом імітації реальних атак. У цій статті розглядається застосування мультиагентних систем (MAS) для автоматизації процесу пентестингу, зокрема для виявлення XSS-вразливостей.

Пентестинг є важливим етапом забезпечення безпеки вебзастосунків, який включає кілька етапів: початковий аналіз, виявлення вразливостей, експлуатацію вразливостей і оцінку наслідків атаки. Типові вразливості, зокрема XSS, є одними з основних цілей пентестингу, оскільки їх можна легко використовувати для компрометації системи. Однак традиційні методи пентестингу часто мають обмежену здатність швидко адаптуватися до нових атак, що робить їх менш ефективними.

З огляду на ці обмеження у статті запропоновано використання мультиагентних систем для автоматизації пентестингу. Такий підхід дозволяє паралельно виконувати різні завдання, що значно збільшує ефективність процесу. Розроблений агент-сканер для виявлення XSS-вразливостей протестовано на вразливому вебресурсі, що дозволило оцінити ефективність цієї моделі. Результати показали, що мультиагентний підхід дозволяє швидше і точніше виявляти вразливості порівняно з традиційними методами. Крім того, цей підхід забезпечує високу гнучкість та адаптивність до нових кіберзагроз.

Загалом, дослідження підтвердило, що використання мультиагентних систем може значно підвищити ефективність пентестингу вебзастосунків. Подальші дослідження можуть бути спрямовані на розширення можливостей таких систем шляхом інтеграції нових агентів для виявлення інших типів вразливостей та використання методів машинного навчання для адаптації агентів до нових загроз.

Ключові слова: *мультиагентні системи, пентестинг, XSS-вразливості, автоматизоване виявлення, веббезпека.*

Вступ

З розвитком інформаційних технологій та зростанням обсягів обробки чутливої інформації в Інтернеті вебзастосунки стали критично важливою частиною сучасних бізнес-процесів. Водночас зростаючий рівень кіберзагроз вимагає від організацій впровадження ефективних методів забезпечення безпеки своїх вебресурсів. Одним з основних інструментів забезпечення безпеки вебзастосунків є тестування на проникнення, яке передбачає виявлення вразливостей шляхом імітації реальних атак. У цій статті досліджується використання мультиагентних систем (MAS) для автоматизації процесу пентестингу, зокрема для виявлення однієї з найбільш поширених типових вразливостей – XSS (Cross-Site Scripting).

Пентестинг є невід'ємною частиною процесу забезпечення безпеки вебзастосунків, оскільки дозволяє виявити потенційні вразливості, які можуть

ТЗ. №4(36)’2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

бути використані зловмисниками для несанкціонованого доступу до даних або інших шкідливих дій. Тестування включає кілька етапів: початковий аналіз, виявлення вразливостей, експлуатацію цих вразливостей, а також подальше оцінювання можливих наслідків атаки. Типові вразливості, такі як SQL-ін’єкції, XSS та CSRF (Cross-Site Request Forgery), є одними з основних цілей пентестингу, оскільки вони можуть бути легко використані для компрометації системи. Тому ефективні інструменти для автоматизації цього процесу, такі як мультиагентні системи, набувають усе більшого значення.

Однією з головних проблем традиційних методів пентестингу є їх обмежена здатність швидко адаптуватися до нових умов та варіантів атак, що постійно змінюються. Для подолання цих обмежень у статті пропонується використання мультиагентних систем, здатних автоматизувати процес тестування на проникнення через взаємодію різних агентів, що виконують різні функції. У рамках дослідження розроблено спеціалізованого агента для пошуку XSS-вразливостей, який протестований на спеціально створеному вразливому вебресурсі, що дозволило оцінити ефективність цієї моделі. Завдяки мультиагентному підходу досягнуто більшої точності та швидкості виявлення вразливостей порівняно з традиційними методами. Це відкриває нові перспективи для використання MAS у пентестингу вебзастосунків, забезпечуючи більш гнучкий та ефективний підхід до тестування безпеки вебресурсів.

Аналіз літературних джерел та постановка проблеми

У статті [1] пропонується огляд різних агентських платформ, на яких здійснюються агентно-орієнтовані симуляції, з акцентом на важливість вибору правильної платформи для створення адаптивних та ефективних агентних систем. Це дослідження висвітлює як теоретичні принципи, так і практичні аспекти використання таких платформ у різних галузях. Окремо у дослідженні [2] підкреслюється важливість агентів у прогнозуванні складних систем та прийнятті рішень, що робить це дослідження важливим для подальшого розвитку теорії та практики агентних технологій.

Дослідження [3] зосереджене на розгляді агентно-орієнтованих платформ, надаючи рекомендації щодо їх розробки та вдосконалення, і визначає проблеми, що виникають під час створення масштабованих агентних систем. У цьому контексті стаття [4] розглядає платформи для розробки агентних систем, звертаючи увагу на їхню роль у вирішенні складних аналітичних задач, зокрема, підкреслюючи важливість правильного вибору платформи для досягнення успіху.

Манева у роботі [5] досліджує архітектуру платформ для агентного аналізу бізнес-процесів, що є важливою темою в контексті автоматизації та оптимізації бізнес-процесів через використання агентних систем. З іншого боку, дослідження [6] присвячене сучасним програмним засобам для агентно-орієнтованого моделювання та можливостям, які надають нові технології для створення ефективних і гнучких агентних платформ.

Стаття [7] зосереджена на оцінці агентної платформи, розробленої на основі мови програмування JAVA, що дозволяє зрозуміти важливі аспекти інтеграції таких платформ у вже наявні програмні середовища. Тоді як у статті [8] пропонується концепція мобільних агентів і платформ для їх реалізації, що є важливим кроком для розвитку мобільних агентних систем.

У дослідженні [9] представлений підхід до теорії динаміки платформ, що має велике значення для глибшого розуміння взаємодії між різними платформами в екосистемах агентів, що дозволяє створювати інтегровані та ефективні рішення для широкого спектра завдань. В роботі [10] пропонується аналіз реальних умов використання мобільних агентів, що є важливим для впровадження мобільних агентних технологій у різних сферах.

У статті [11] обговорюється адаптивне формування мультиагентних систем, що піддаються атакам типу denial-of-service, а також підхід до забезпечення їх безпеки і стійкості. Це пов’язано з дослідженням [12], яке стосується стратегічної логіки для аналізу динамічних моделей, де описується новий підхід до моделювання взаємодії агентів в умовах зміни середовища, що може бути застосовано для оптимізації процесів.

Важливою темою також є питання безпеки в мультиагентних системах, яке висвітлюється у статті [13]. Огляд різних загроз та вразливостей у таких системах підкреслює значення розробки методів захисту. Для забезпечення безпеки і стійкості мультиагентних систем у складних умовах розглядається консенсусне управління у роботі [14], де запропоновано стійкі нейроадаптивні підходи для кібербезпеки.

Дослідження [15] пропонує огляд застосування мультиагентних систем у різних галузях, таких як Інтернет речей і кіберфізичні системи, що є важливим для розуміння потенціалу таких систем у розвитку технологій майбутнього. Водночас у роботі [16] пропонується модель кіберфізичної системи на основі мультиагентної архітектури для оптимізації взаємодії компонентів у таких системах, що має велике значення для розвитку інтелектуальних кіберфізичних систем.

Дослідження [17] охоплює застосування мультиагентних систем для контролю мікрогрід, з аналізом підходів для ефективного управління енергетичними системами в умовах змінної навантаженості. Своєю чергою дослідження [18] розглядає MAS-ML 2.0,

ТЗ. №4(36)’2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

систему для моделювання мультиагентних систем з різними архітектурами агентів, що дозволяє досягти більшої гнучкості та ефективності в таких системах.

Стаття [19] зосереджена на протоколах переговорів для мультиагентних систем, що використовуються у кіберфізичних системах, підкреслюючи важливість ефективної комунікації між агентами. Окремо у статті [20] досліджується роль мультиагентних систем у розподілі ресурсів і плануванні в розумних мережах, що важливо для оптимізації енергетичних ресурсів у змінних умовах.

У статті [21] розглядається виявлення вразливостей у вебзастосунках за допомогою пентестингу, підкреслюючи важливість застосування методів і інструментів для перевірки безпеки вебзастосунків. Останні дослідження в цій сфері, наприклад, у роботі [22], пропонують удосконалення інструментів для тестування безпеки, а в роботі [23] досліджуються проблеми автоматизованого тестування вразливостей.

Інтерес до покращення інструментів пентестингу також висвітлюється у статті [24], де розглядається вебінтерфейс для моніторингу діяльності пентесту на основі стандартів OWASP, що сприяє ефективному управлінню процесом тестування безпеки вебзастосунків. Важливими є також дослідження [25] і [26], де аналізуються проблеми використання чорних сканерів у пентестингу та інтеграція результатів OpenVAS у Metasploit для підвищення ефективності тестування.

У статті [27] наводиться емпіричне порівняння інструментів пентестингу для виявлення вразливостей вебзастосунків, що є важливим для вибору найефективніших інструментів для тестування безпеки вебсистем. Дослідження [28] присвячене проблемам безпеки вебзастосунків та оцінці вразливостей через пентестинг, розглядаючи загальні питання безпеки та методи їх тестування для забезпечення захисту вебресурсів.

У статті [29] детально описано пентестинг вебзастосунків, включаючи основні стратегії та інструменти, що використовуються для виявлення вразливостей. Нарешті, в роботі [30] розглядаються нові виклики у сучасному пентестингу, зокрема виявлення нових типів вразливостей та застосування новітніх технологій для тестування безпеки.

Аналіз джерел свідчить, що мультиагентні системи мають значний потенціал у сфері автоматизації тестування безпеки, оскільки вони дозволяють здійснювати паралельну обробку завдань та адаптуватися до нових загроз. У дослідженнях [15; 16; 18] розглядаються можливості застосування мультиагентних систем у кіберфізичних системах та Інтернеті речей, що підтверджує їхню ефективність у динамічних і складних середовищах.

Постановка проблеми дослідження

Вебзастосунки є основними елементами сучасних інформаційних систем, використовуваних у різноманітних сферах – від бізнесу до урядових і фінансових структур. Водночас вони стали головною мішенню для кіберзлочинців, які шукають можливості для несанкціонованого доступу до чутливої інформації, компрометації систем або викрадення даних. Кількість і складність кіберзагроз постійно зростає, а нові типи вразливостей з’являються регулярно. Такі атаки, як XSS, SQL-ін’єкції, CSRF і багато інших, мають потенціал для серйозних порушень безпеки вебзастосунків. З огляду на високі ризики необхідність забезпечення надійної безпеки вебресурсів стає дедалі актуальнішою для організацій і державних установ.

Традиційні методи пентестингу часто потребують значних людських ресурсів і часу, що може обмежувати їхню ефективність і швидкість. Аналіз літературних джерел, зокрема досліджень [21–24], демонструє актуальність автоматизації процесу тестування безпеки вебзастосунків для оперативного виявлення вразливостей. Використання мультиагентних систем, як це показано у роботах [15–18], дозволяє суттєво підвищити ефективність та гнучкість процесу пентестингу завдяки можливості паралельного виконання завдань і швидкої адаптації до нових загроз.

З огляду на вищезазначене мета цього дослідження визначається як розробка та впровадження мультиагентної системи для автоматизації пентестингу вебзастосунків, зокрема для пошуку вразливостей типу XSS. Розробка такого підходу дозволить значно підвищити ефективність і швидкість виявлення вразливостей, а також зробити процес тестування більш гнучким і адаптивним до нових кіберзагроз.

Основними завданнями, які розв’язуються в цьому дослідженні, є:

– аналіз та визначення основних етапів пентестингу вебзастосунків, зокрема етапу пошуку XSS-вразливостей;

– використання мультиагентної системи для автоматизації процесу пентестингу з урахуванням специфіки вебзастосунків;

– створення спеціалізованого агента-сканера для пошуку XSS-вразливостей у вебресурсах;

– тестування розробленої системи на спеціально створеному вразливому вебресурсі, оцінка ефективності та точності виявлення вразливостей;

– оцінка результатів і порівняння ефективності агента-сканера з аналогічними системами виявлення вразливостей.

ТЗ. №4(36)’2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

Матеріали та методи досліджень

Для проведення дослідження використані комбінації фундаментальних та спеціальних підходів до тестування на проникнення вебзастосунків з використанням мультиагентних систем. Основними методами стали аналіз вразливостей вебзастосунків, таких як XSS, метод MAS для створення агентів, що виконують тестування, а також автоматизація процесу пентестингу для підвищення ефективності. Емпіричний метод дозволив оцінити розроблені моделі тестування на вразливих вебресурсах.

Для розробки агента-сканера використана мова програмування Python. Python забезпечує гнучкість і швидкість завдяки бібліотекам для вебтестування, таким як Asyncio, OS, Httpx, Requests. Як платформа MAS для створення та організації взаємодії агентів (crawler-a та scanner-a) всередині нашої моделі виступає SPADE (Smart Python Agent Development Environment) [38]. Для оцінки ефективності агентної системи використано вразливий вебресурс (vulnweb.com), на якому тестувалися агенти. Всі етапи тестування організовані за допомогою систем управління проєктами, таких як Jira та Trello, що дозволило ефективно координувати дослідження та збір результатів. Використання цих інструментів та технологій дозволило досягти поставленої мети щодо автоматизації процесів виявлення вразливостей типу XSS у вебзастосунках.

У цьому дослідженні, як відзначалось раніше, SPADE використовується для побудови системи взаємодії двох агентів, забезпечуючи асинхронну взаємодію між ними через обмін повідомленнями по протоколу XMPP. SPADE допомагає в організації незалежних процесів, коли кожен агент виконує свою роботу автономно. Також дозволяє агентам працювати асинхронно без необхідності використовувати додаткові черги або бази даних (Redis, RabbitMQ тощо). Також SPADE дає можливість використання вебінтерфейсу для моніторингу стану агентів у реальному часі.

У SPADE поведінка агентів задається через спеціальні класи поведінки (behaviours). SPADE підтримує три основні типи поведінки:

– OneShotBehaviour – виконується один раз і завершується.

– CyclicBehaviour – виконується безкінечно або до зупинки агента.

– PeriodicBehaviour – виконується через певний інтервал часу.

У коді, який запропонований у цьому дослідженні, використовується OneShotBehaviour.

Результати дослідження

Результатом проведеного дослідження є розроблений агент-сканер, який автоматизує процес виявлення вразливостей типу XSS, зокрема Reflected та DOM Based, у знайдених URL цільового домену. Запропонований підхід є особливо корисним у випадках, коли необхідно перевірити значний обсяг доменів або вебресурс із великою кількістю кінцевих точок, що можуть містити вразливі до XSS параметри. Використання такого інструменту дозволяє значно підвищити якість та повноту перевірки всього досліджуваного середовища, а також зменшити витрати часу на ручний аналіз кожного окремого параметра.

Розроблений сканер вимагає від користувача лише вказання цільового домену, після чого всі процеси виконуються в автоматизованому режимі. Отримані результати надсилаються користувачеві для подальшої валідації, що дозволяє ефективно ідентифікувати потенційні вразливості та здійснити їх аналіз.

Життєвий цикл роботи розробленої моделі представлено на рис. 1.

На початковому етапі тестувальник передає агенту-сканеру цільовий домен у вигляді URL-адреси, наприклад, https://example.com. Перший агент (crawler_agent) отримує цю адресу та ініціює активний процес збору URL, пов’язаних із цільовим доменом. Зібрані результати підлягають фільтрації з метою виокремлення URL, що містять параметри. Визначення таких URL здійснюється шляхом аналізу наявності у рядку символів ?, &, =, наприклад, https://example .com?test=123&case=456. У цьому випадку параметрами є test і case, а значеннями цих параметрів відповідно 123 та 456. Подальший

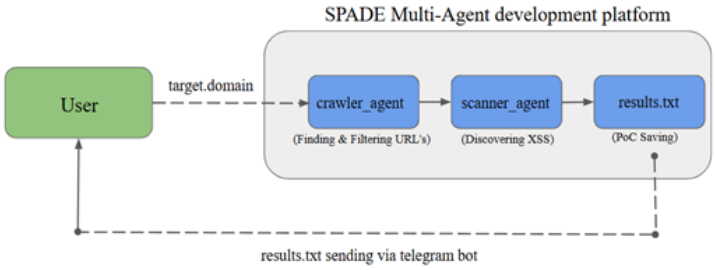

Рис. 1. Життєвий цикл роботи розробленої моделі

ТЗ. №4(36)’2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

етап передбачає інжекцію спеціально сформованих корисних навантажень (пейлоудів) для перевірки можливих XSS-вразливостей. Для цього можна використати такі приклади:

https://example.com?test=<svg/onload=alert(1)>&case=456

https://example.com?test=»><script>alert(1)</script>&case=456

https://example.com?test=123&case=4»><img src=x oneerror=alert(1)

https://example.com?test=123&case=\’-alert(1)//

Якщо в процесі аналізу відповідь сервера містить впроваджене корисне навантаження, і після переходу за відповідним посиланням викликається функція alert() у вигляді спливаючого вікна, це свідчить про вразливість вебресурсу до XSS-атак.

Після етапів збору, фільтрації та ідентифікації потенційно вразливих параметрів результати передаються наступному агенту (scanner_agent), який здійснює безпосередню перевірку URL-адреси на наявність XSS-вразливостей. Виявлені вразливості зберігаються у вигляді Proof-of-Concept (PoC) у текстовий файл. Далі ці результати передаються користувачеві через телеграм-бота, створеного за допомогою BotFather [31].

Під час розробки агента-сканера використовувалися наявні утиліти з відкритим вихідним кодом: Katana [32], Gxss [33] і Dalfox [34]. Вказані інструменти послідовно запускаються, утворюючи автоматизований ланцюг аналізу.

Katana є високопродуктивним та гнучко налаштовуваним вебкраулером від ProjectDiscovery, призначеним для виявлення всіх можливих URL-адрес на вебсайті, включаючи динамічні сторінки, форми та API-запити. Утиліта реалізує рекурсивний пошук, аналізує HTML-код, JavaScript, витягує вбудовані скрипти, API-запити та параметри.

Gxss виконує аналіз знайдених URL-адрес на наявність параметрів, потенційно вразливих до XSS-атак. Цей інструмент обробляє результати, отримані за допомогою Katana, визначає параметри, що можуть слугувати точками ін’єкцій, та здійснює їх фільтрацію, залишаючи лише ті, які можуть бути об’єктами атаки.

Dalfox є комплексним інструментом для виявлення та валідації XSS-вразливостей у знайдених точках ін’єкцій. Він здійснює інжекцію XSS-пейлоудів у параметри, аналізує відповідь сервера на предмет виконання JavaScript-коду, а також оцінює різні контексти виконання (HTML, JS, атрибути, події). Крім того, Dalfox містить вбудовані алгоритми штучного інтелекту для обходу захисних механізмів, зокрема політики безпеки контенту (CSP, Content-Security Policy) [35] і фаєрволів вебзастосунків (WAF, Web-Application Firewall) [36].

Комбінація зазначених утиліт формує повноцінний автоматизований сканер для виявлення XSS-вразливостей. Загальна схема його роботи виглядає таким чином:

katana -u https://example.com -silent | grep -E «\?|&=» | Gxss -c 100 | dalfox pipe

Демонстрація роботи розробленої моделі. Автоматизація процесу пошуку XSS-вразливостей є ключовим завданням у сфері кібербезпеки, оскільки ручна перевірка великої кількості вебресурсів є трудомісткою та неефективною. Запропонована модель агента-сканера дозволяє значно підвищити швидкість та точність виявлення потенційних загроз. У цьому розділі представлено демонстрацію роботи розробленого сканера, процес збору URL-адрес, пошук потенційних точок ін’єкції, а також аналіз отриманих результатів.

Розроблена модель агента-сканера функціонує за принципом автоматизованого пошуку вразливостей XSS у вебзастосунках. На початковому етапі користувач задає цільовий домен, після чого виконується процес збору URL-адрес (див. рис. 2).

Після збору URL-адрес виконується їх аналіз для виявлення потенційних точок ін’єкції за допомогою Gxss (див. рис. 3).

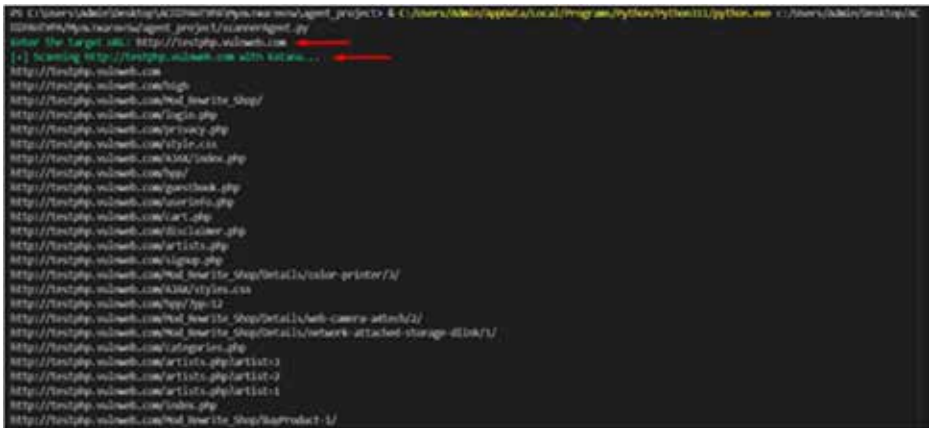

Рис. 2. Початок роботи моделі. Збір URL-адрес

Т3. №4(36)’2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

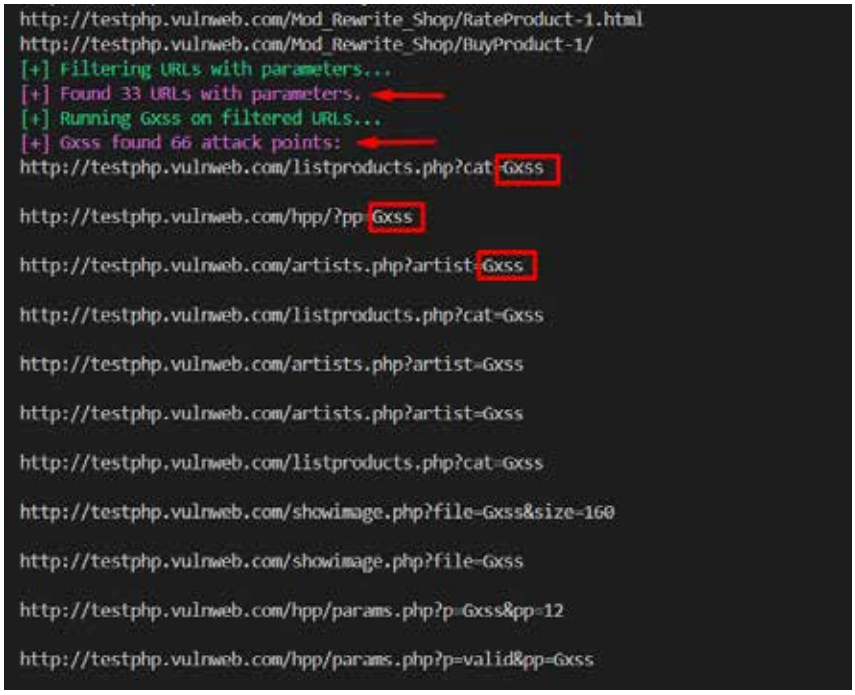

Рис. 3. Пошук точок для атаки

Наступним етапом є передача отриманих URL-адрес до модуля dalfox для виявлення XSS-вразливостей (див. рис. 4).

Результати роботи сканера представлено на рис. 5.

Підсумкові дані зберігаються у текстовий файл та надсилаються користувачу через Telegram-бот (див. рис. 6–7).

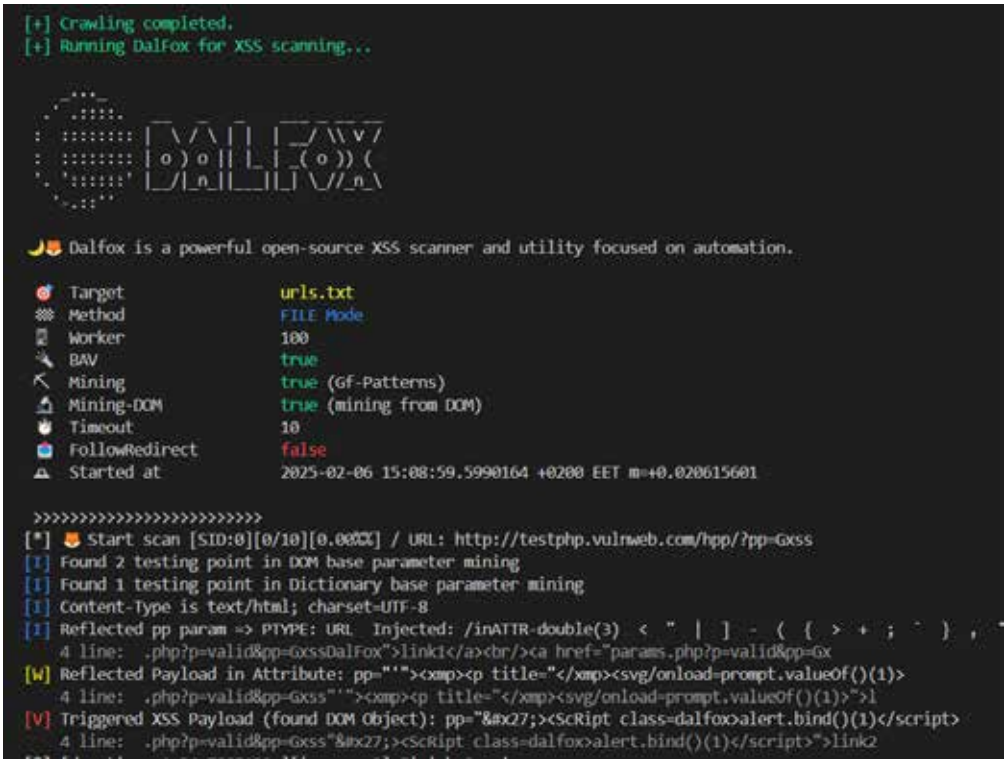

Рис. 4. Початок сканування на XSS

ТЗ. №4(36)’2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

Рис. 5. Результат роботи сканера

Користувач отримує перелік URL-адрес із вбудованими корисними навантаженнями у текстовому форматі. Після цього виконується їх ручна перевірка для підтвердження статусу true-positive. Для цього необхідно скопіювати отриманий результат та відкрити відповідне посилання у браузері.

Для підтвердження вразливості виконуються маніпуляції з переданими пейлоудами (див. рис. 8–10).

Приклад 1:

http://testphp.vulnweb.com/hpp/?pp=Gxss%22%27%22%3E%3Cxmp%3E%3Cp+title%3D%22%3C%2Fxmp%3E%3Csvg%2Fonload%3Dprompt.valueOf%28%29%281%29%3E

URL-декодована форма: http://testphp.vulnweb.com/hpp/?pp=Gxss»‘«><xmp><p+title=»</xmp><svg/onload=prompt.valueOf()(1)>

У цьому прикладі:

Gxss» закриває атрибут pp;

>‘ завершує поточний HTML-тег, дозволяючи вставити новий HTML-код;

<xmp> – це HTML-тег, який відображає вміст як текст, але його можна закрити (</xmp>), щоб вставити наступний код;

<svg onload=prompt.valueOf()(1)> створює SVG-елемент, який у разі завантаження викликає prompt.valueOf()(1), що функціонально еквівалентне виклику alert(1).

Приклад 2:

http://testphp.vulnweb.com/hpp/params.php?p=Gxss&pp=12%3CsVg%2Fonload%3Dalert%281%29%3E

URL-декодована форма:

http://testphp.vulnweb.com/hpp/params.php?p=Gxss&pp=12<sVg/onload=alert(1)>

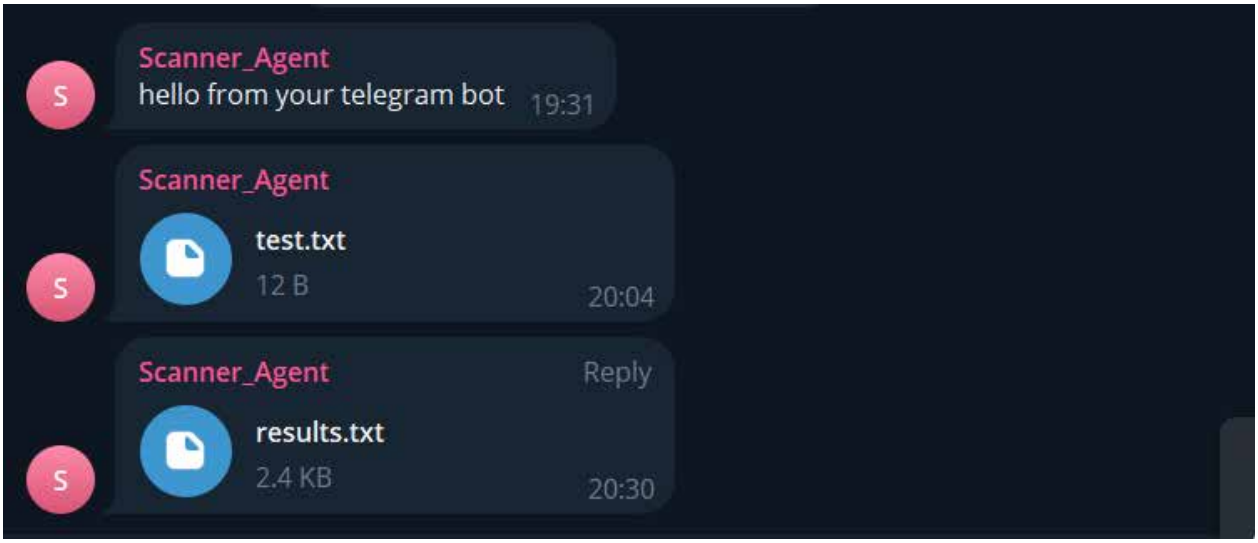

Рис. 6. Надіслані результати у Telegram

ТЗ. №4(36)’2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

Рис. 7. Вміст файлу із результатами

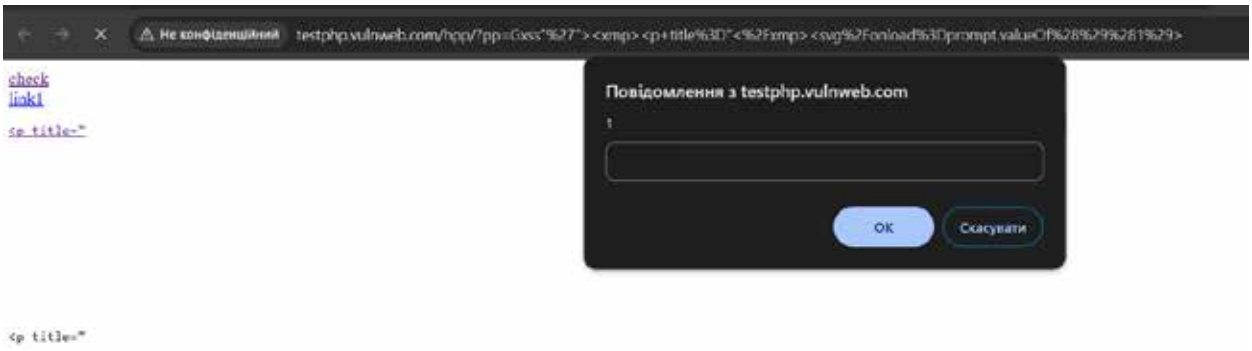

Рис. 8. Приклад 1

У прикладі 12<svg onload=alert(1)> вставляється в HTML-код без екранування, що дозволяє виконати XSS-атаку. <svg onload=alert(1)> створює SVG-елемент, який під час завантаження викликає alert(1), запускаючи JavaScript-код.

Приклад 3:

http://testphp.vulnweb.com/hpp/params.php?p=%3Cimg%2Fsrc%2Fonerror%3D.1%7Calert%601%60%3E&pp=Gxss

URL-декодована форма:

http://testphp.vulnweb.com/hpp/params.php?p=<img/src/onerror=.1|alert`1`>&pp=Gxss

У прикладі <img src onerror=.1|alert1> створює зображення з некоректним src, що змушує браузер викликати oneerror. onerror=.1|alert1`` у разі помилки завантаження спочатку виконує .1, а потім викликає alert(1), запустивши JavaScript.

Використання бектіків (`) та | допомагає обійти деякі фільтри XSS.

Запропонований сканер дозволяє виявляти XSS-вразливості шляхом виконання JavaScript-коду на стороні клієнта, що підтверджує можливість експлуатації вразливості у вебзастосунку.

Незважаючи на ефективність розробленого сканера, наявні можливості його подальшого покращення:

– Гнучкість налаштувань – реалізація підтримки кількох доменів, користувацьких XSS-пейлоудів, а також інтеграція proxy/VPN.

– Розширення збору URL-адрес – використання пасивних методів збору, таких як gau та waybackurls, для отримання архівних та прихованих сторінок.

– Виявлення прихованих параметрів – інтеграція ParamSpider та Arjun для ідентифікації прихованих точок ін’єкції.

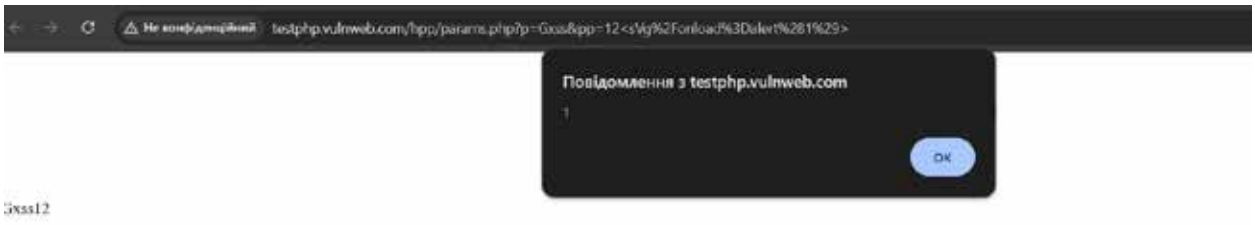

Рис. 9. Приклад 2

ТЗ. №4(36)’2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

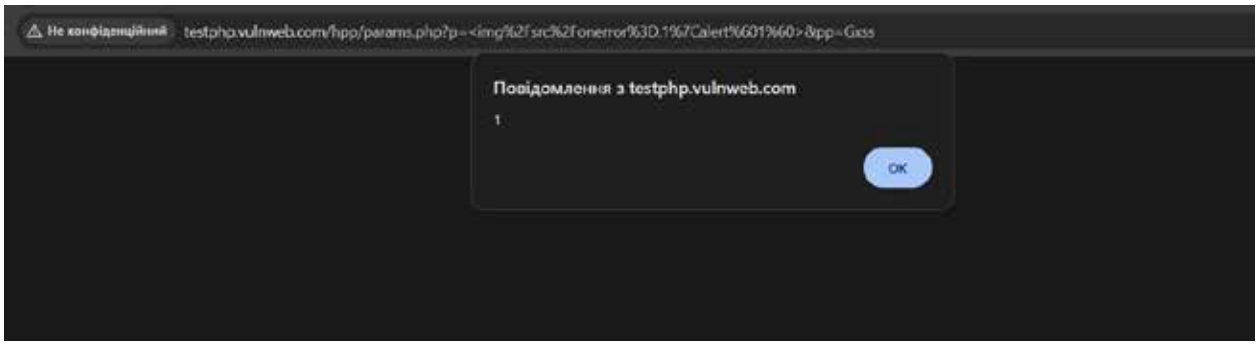

Рис. 10. Приклад 3

– Евристичний аналіз точок ін’єкції – аналіз стандартних параметрів (q=, search=, id=, page=) для підвищення точності виявлення XSS.

– Обхід WAF та обмежень – застосування wafw00f для визначення WAF та механізмів обходу обмежень (зміна заголовків, обфускація пейлоудів, ротація IP-адрес через Tor або проксі).

– Оптимізація продуктивності – використання багатопоточного режиму Dalfox та масштабування процесу сканування за допомогою asyncio.create_subprocess_exec().

– Автоматизація сповіщень – інтеграція зі службами Telegram, Discord, Slack для оперативного отримання інформації про знайдені вразливості.

– Розширення покриття тестування – паралельний запуск сканування на SQL Injection (SQLMap), SSRF, CSRF, Open Redirects та інші типи вразливостей.

Порівняння зі сканером Burp Suite professional. Burp Suite [37] є комплексним інструментом для оцінки безпеки вебзастосунків, який визнаний одним з найбільш потужних у своєму класі. За результатами опитувань серед пентестерів з усього світу 94% з них вважають Burp Suite найкращим інструментом для тестування безпеки. Такий інструмент дозволяє виконувати практично всі необхідні операції для повноцінного тестування на проникнення. У складі Burp Suite є як вбудований сканер для виявлення вразливостей, так і набір інструментів для ручного тестування. Проте варто зазначити, що доступ до сканера можливий лише у платній версії, вартість якої на поточний момент становить 449 доларів на рік.

Далі представлене порівняння ефективності нашого сканера з інструментом сканування Burp Suite. Для цього проведемо сканування на вразливість XSS на одному й тому ж цільовому домені (див. рис. 11–13).

Нижче представлено класичне відображення ідентифікованих вразливостей у програмному інтерфейсі Burp Suite. Для нашого прикладу знайдено 8 вразливих параметрів (див. рис. 11).

Приклад ручної перевірки окремого запиту та відповіді від сервера з успішно інжектованим пейлоудом у параметр *uuname* демонструє наявність вразливості до атак типу XSS. Це підтверджується наявністю успішно відкритого та закритого тегу `<script>`, що містить виклик функції `alert()` в HTML-коді (див. рис. 12).

Демонстрацію відпрацювання пейлоуду представлено на рис. 13.

У результаті проведеного тестування за допомогою Burp Suite було виявлено 8 вразливостей типу XSS. Сканування здійснювалося на вебресурсі http://testphp.vulnweb.com. Далі буде проведено порівняння ключових функціональних характеристик вибраних інструментів для оцінки їх ефективності.

З огляду на дані, наведені в таблиці, можна зробити висновок щодо ефективності розробленого агента-сканера порівняно з просунутим комерційним сканером Burp Suite. Перевагами агента-сканера

Time	Source	Issue type	Host	Path	Insertion point	Severity	Confidence
17:15...	Task 9	Cross-site scripting (reflected)	http://testphp.vulnweb.com	/secured/newuser.php	uaddress parameter	High	Certain
17:15...	Task 9	Cross-site scripting (reflected)	http://testphp.vulnweb.com	/secured/newuser.php	uphone parameter	High	Certain
17:11...	Task 9	Cross-site scripting (reflected)	http://testphp.vulnweb.com	/hpp/	pp parameter	High	Certain
17:05...	Task 9	Cross-site scripting (reflected)	http://testphp.vulnweb.com	/hpp/params.php	p parameter	High	Certain
17:05...	Task 9	Cross-site scripting (reflected)	http://testphp.vulnweb.com	/secured/newuser.php	uemail parameter	High	Certain
17:05...	Task 9	Cross-site scripting (reflected)	http://testphp.vulnweb.com	/secured/newuser.php	ucc parameter	High	Certain
17:05...	Task 9	Cross-site scripting (reflected)	http://testphp.vulnweb.com	/secured/newuser.php	urname parameter	High	Certain
17:04...	Task 9	Cross-site scripting (reflected)	http://testphp.vulnweb.com	/secured/newuser.php	uuname parameter	High	Certain
17:12...	Task 9	Cross-site scripting (reflected)	http://testphp.vulnweb.com	/showimage.php	file parameter	Information	Certain

Рис. 11. Знайдені вразливості сканером Burp Suite

ТЗ. №4(36)’2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

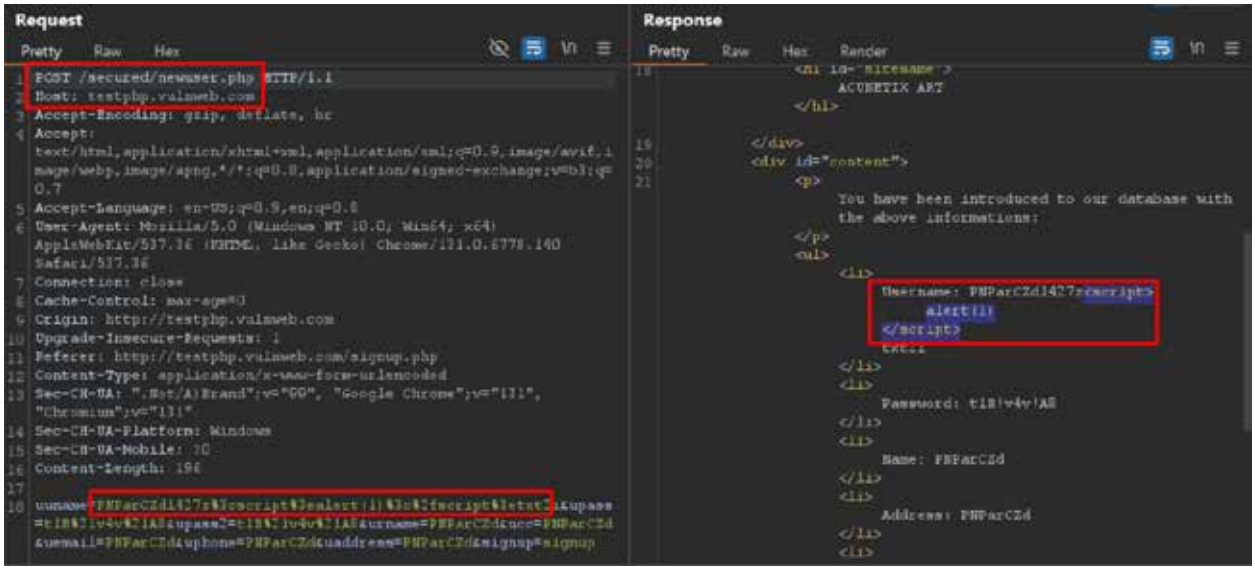

Рис. 12. Приклад запиту та відповіді із PoC

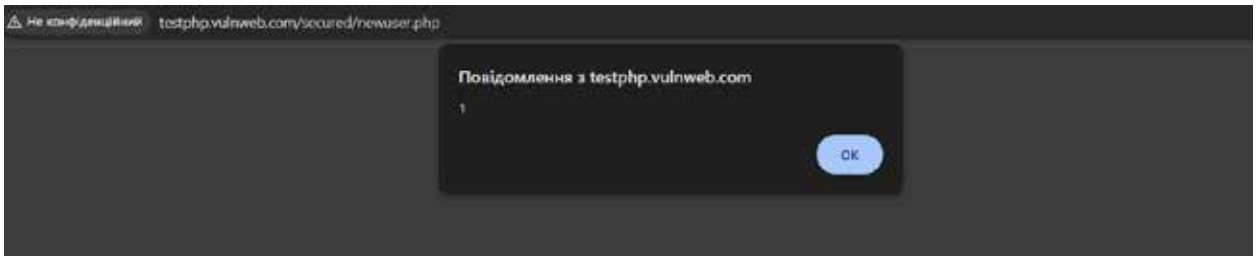

Рис. 13. Відпрацювання функції alert

Табл. 1. Порівняння агента-сканера та сканера Burp Suite

Критерій	Агент-сканер	Burp Suite сканер
Вартість	безкоштовно	449 $ на рік
Кількість ідентифікованих вразливих параметрів	6	8
Типи XSS	Reflected, DOM Based, Stored	Reflected, DOM Based, Stored
Вбудований краулер	Так	Так
Час краулінгу	< 1 хв	< 5 хв
Час сканування	< 5 хв	> 11 хв
True-positive	80–90%	85–95%
Зручність користування	Зручний	Зручний
Обхід захисних механізмів	Автоматичний	Із доналаштуваннями
Звітування	Тільки PoC	Повний звіт
Масштабованість	Висока, оптимізований для обробки великих обсягів даних за допомогою SPADE	Висока, здатен працювати з великими вебзастосунками, потребує більших апаратних ресурсів
Продуктивність на великих вебзастосунках	Висока продуктивність, оптимізовано для великих вебзастосунків	Висока продуктивність у разі налаштування та оптимізації

ТЗ. №4(36)'2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

є його швидкість, здатність до обходу захисних механізмів та відсутність вартості. Обидва сканери виявили вразливі параметри, які не були знайдені іншим інструментом, що свідчить про їхню доповнювальність. Так, Burp Suite виявив вразливість, яку не знайшов агент-сканер, і навпаки. Отже, для досягнення більш ефективних результатів під час тестувань доцільно використовувати їх разом за можливості. Для отримання більш точних результатів необхідне подальше глибоке дослідження на різних цільових доменах.

Висновки

У результаті проведеного дослідження був розроблений агент-сканер для тестування на проникнення вебзастосунків, що включає автоматизоване виявлення вразливостей типу XSS. Модель агента успішно протестована на вразливому вебресурсі, що дозволило оцінити її ефективність у реальних умовах та порівняти з просунутим комерційним аналогом Burp Suite.

Ефективність запропонованого підходу підтверджена результатами сканування, які продемонстрували переваги агента-сканера, серед яких –швидкість роботи, здатність обходити захисні механізми та відсутність вартості. Мультиагентна система забезпечила високу гнучкість у процесі створення робочої моделі та дозволила автоматизувати окремі етапи пентестингу, що значно скоротило час, необхідний для проведення перевірок безпеки.

Важливим результатом є можливість подальшого вдосконалення цієї технології через інтеграцію нових агентів для виявлення інших типів вразливостей, а також застосування машинного навчання для адаптації агентів до нових загроз. Подальші дослідження можуть бути спрямовані на розширення застосування мультиагентних систем у пентестингу та їх інтеграцію з іншими методами захисту вебзастосунків, такими як аналіз поведінки користувачів та інтелектуальні системи попередження атак.

Конфлікт інтересів

Автори декларують, що не мають конфлікту інтересів стосовно цього дослідження, в тому числі фінансового, особистісного характеру, авторства чи іншого характеру, що міг би вплинути на дослідження та його результати, представлені в цій статті.

Фінансування

Дослідження проводилося без фінансової підтримки.

Доступність даних

Рукопис не має пов'язаних даних.

ЛІТЕРАТУРА

[1] Kravari K. and Bassiliades N. A survey of agent platforms. J. Artif. Soc. Soc. Simul., vol. 18, 2015. DOI: 10.18564/jasss.2661.

[2] Gilbert N. and Bankes S. Platforms and methods for agent-based modeling. Proc. Natl. Acad. Sci. U. S. A., vol. 99, pp. 7197–7198, May 2002. DOI: 10.1073/PNAS.072079499.

[3] Railsback S., Lytinen S., and Jackson S. Agent-based simulation platforms: Review and development recommendations. Simulation, vol. 82, pp. 609–623, Sep. 2006. DOI: 10.1177/0037549706073695.

[4] Pal C.-V., Leon F., Paprzycki M., and Ganzha M. A review of platforms for the development of agent systems. Inf., vol. 14, p. 348, Jul. 2020. DOI: 10.3390/info14060348.

[5] Maneva R. Development of agent platform architecture for intelligent analysis of business processes. Bionics of Intelligence, 2020. DOI: 10.30837/bi.2020.1(94).09.

[6] Bragin A. Modern software tools for agent-based modeling. Artificial Societies, 2022. DOI: 10.18254/s207751800023501-0.

[7] Vrba P. JAVA-based agent platform evaluation. Sep. 2003, pp. 47–58. DOI: 10.1007/978-3-540-45185-3_5.

[8] Puliafito A., Tomarchio O., and Vita L. MAP: Design and implementation of a mobile agents' platform. J. Syst. Archit., vol. 46, pp. 145–162, Jan. 2000. DOI: 10.1016/S1383-7621(98)00076-9.

[9] Cabral L. Towards a theory of platform dynamics. ERN: Other Organizations & Markets: Personnel Management (Topic), Jul. 2018. DOI: 10.1111/jems.12312.

[10] Altmann D. and Gruber A. Using mobile agents in real world: A survey and comparison of different approaches. Semantic Scholar. [Online]. URL: https://surl.gd/tijhjn.

[11] Pan K., Lyu Y., and Pan Q. Adaptive formation for multiagent systems subject to denial-of-service attacks. IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, pp. 3391–3401, Aug. 2022. DOI: 10.1109/TCSI.2022. 3168163.

[12] Catta D., Leneutre J., Malvone V., and Murano A. Obstruction alternating-time temporal logic: A strategic logic to reason about dynamic models. pp. 271–280, 2024. DOI: 10.5555/3635637.3662875.

[13] Cavalcante R., Bittencourt I., Silva A., Silva M., Costa E., and Santos R. A survey of security in multi-agent systems. Expert Systems with Applications, vol. 39, pp. 4835–4846, Apr. 2012. DOI: 10.1016/j.eswa.2011.09.130.

[14] Yang X., Yang L., Dong L., Jin W.-H., Zhang M., Yang F., and Lin Y. Consensus tracking control for uncertain non-strict feedback multi-agent system under cyber-attack via resilient neuroadaptive approach. International Journal of Robust and Nonlinear Control, vol. 32, pp. 4251–4280, Feb. 2022. DOI: 10.1002/rnc.6035.

[15] Dorri A., Kanhere S., and Jurdak R. Multi-agent systems: A survey. IEEE Access, vol. 6, pp. 28573–28593, Apr. 2018. DOI: 10.1109/ACCESS.2018.2831228.
[16] Ahmed M., Kazar O., and Harous S. Cyber-physical system model based on multi-agent system. IET Cyber-Physical Systems: Theory & Applications, Jun. 2024. DOI: 10.1049/cps2.12096.
[17] Kantamneni A., Brown L., Parker G., and Weaver W. Survey of multi-agent systems for microgrid control. Engineering Applications of Artificial Intelligence, vol. 45, pp. 192–203, Oct. 2015. DOI: 10.1016/ j.engappai.2015.07.005.
[18] Gonçalves E., Cortés M., Campos G., Lopes Y.S., Freire E., Silva V., Oliveira K., and De Oliveira M.A. MAS-ML 2.0: Supporting the modelling of multi-agent systems with different agent architectures. Journal of Systems and Software, vol. 108, pp. 77–109, Oct. 2015. DOI: 10.1016/ j.jss.2015.06.008.
[19] Calvaresi D., Appoggetti K., Lustrissimini L., Marinoni M., Sernani P., Dragoni A., and Schumacher M. Multi-agent systems' negotiation protocols for cyber-physical systems: Results from a systematic literature review. Pp. 224–235, 2018. DOI: 10.5220/0006594802240235.
[20] Binyamin S. and Slama S. Multi-agent systems for resource allocation and scheduling in a smart grid. Sensors, vol. 22, Oct. 2022. DOI: 10.3390/s22218099.
[21] Valentina A., Vishwashri R., and Rajadurai S. Finding Vulnerability in Web Application by using Pentesting. Int. J. Multidiscip. Res., 2024. DOI: 10.36948/ijfmr.2024.v06i04.24517.
[22] Olivares-Naya M., de Gracia J.C., and S'anchez-Maci'an A. Adding web pentesting functionality to PTHelper. ArXiv, vol. abs/2410.12422, 2024. [Online]. URL: https://api.semanticscholar.org/CorpusID:273375081.
[23] De Lima L., Horstmann M., Neto D., Grégio A., Silva F., and Peres L. On the Challenges of Automated Testing of Web Vulnerabilities. In 2020 IEEE 29th Int. Conf. Enabling Technol. Infrastruct. Collaborative Enterprises (WETICE), 2020, pp. 203–206. DOI: 10.1109/WETICE49692.2020.00047.
[24] Wijaya Y. Web-Based Dashboard for Monitoring Penetration Testing Activities Based on OWASP Standards. J. Ilm. Tek. Elektro Komput. Inform., 2020. DOI: 10.26555/jiteki.v16i1.17019.
[25] Doupé A., Cova M., and Vigna G. Why Johnny Can't Pentest: An Analysis of Black-Box Web Vulnerability Scanners. 2010, pp. 111–131. doi: 10.1007/978-3-642-14215-4_7.
[26] Vimala K. and Fugkeaw S. VAPE-BRIDGE: Bridging OpenVAS Results for Automating Metasploit Framework. In 2022 14th Int. Conf. Knowl. Smart Technol. (KST), 2022, pp. 69–74. DOI: 10.1109/KST53302.2022.9729085.
[27] Albahar M., Alansari D., and Jurcut A. An Empirical Comparison of Pen-Testing Tools for Detecting Web App Vulnerabilities. Electronics, 2022. DOI: 10.3390/electronics11192991.
[28] Addressing Web Application Security Issues and Vulnerabilities Assessment Pen Testing. Int. J. Recent Technol. Eng., 2020. DOI: 10.35940/ijrte.f8169.038620.
[29] Al-Ahmad A., Ata B., and Wahbeh A. Pen Testing for Web Applications. Int. J. Inf. Technol. Web Eng., vol. 7, pp. 1–13, 2012. DOI: 10.4018/jitwe.2012070101.
[30] Bertoglio D.D., Gil A., Acosta J., Godoy J., Lunardi R., and Zorzo A. Towards new challenges of modern Pentest. ArXiv, vol. abs/2311.12952, 2023. DOI: 10.48550/arXiv.2311.12952.
[31] Bots: Introduction. Telegram. [Онлайн]. URL: https://surl.gd/ffcfcx. Дата звернення: 20.02.2025.
[32] Discovery P. Katana. GitHub. [Онлайн]. URL: https://surl.gd/tsgljg. Дата звернення: 20.02.2025.
[33] K.P., Gxss. GitHub. [Онлайн]. URL: https://surl.gd/bnuefy. Дата звернення: 20.02.2025.
[34] H. W., Dalfox. GitHub. [Онлайн]. URL: https://surl.gd/plpdtk. Дата звернення: 20.02.2025.
[35] Content Security Policy. [Онлайн]. URL: https://surl.gd/gpihew. Дата звернення: 20.02.2025.
[36] Web application firewall. Wikipedia. [Онлайн]. URL: https://surl.gd/efjsff. Дата звернення: 20.02.2025.
[37] Burp Suite. PortSwigger. [Онлайн]. URL: https://surl.gd/kteluz. Дата звернення: 20.02.2025.
[38] SPADE-MAS. Read the Docs. [Онлайн]. URL: https://surl.gd/uguutq. Дата звернення: 20.02.2025.

AUTOMATED XSS VULNERABILITY DETECTION IN WEB APPLICATIONS BASED ON A MULTI-AGENT APPROACH

Vladyslav Kravchuk, Nataliya Maslova, Iaroslav Dorogyy

With the development of information technologies and the increasing volume of sensitive data processing on the Internet, web applications have become a crucial part of modern business processes. However, with this growth, the number of cyber threats also increases, posing the challenge for organizations to ensure the security of their web resources. One of the primary methods of protection is penetration testing, which allows vulnerabilities to be identified by simulating real attacks. This article explores the use of multi-agent systems (MAS) for automating the penetration testing process, specifically for detecting XSS vulnerabilities.

Penetration testing is a vital step in ensuring the security of web applications, encompassing several stages: initial analysis, vulnerability identification, exploitation of vulnerabilities, and evaluating the consequences of an attack. Common vulnerabilities, such as XSS, are key targets in penetration testing because they can be easily exploited to compromise a system. However, traditional penetration testing methods often have limited ability to adapt quickly to new attacks, making them less effective.

ТЗ. №4(36)'2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

Considering these limitations, this paper proposes the use of multi-agent systems to automate penetration testing. This approach allows different tasks to be executed in parallel, significantly enhancing the effectiveness of the process. A developed agent-scanner for detecting XSS vulnerabilities was tested on a vulnerable web resource, enabling the evaluation of the model's effectiveness. The results showed that the multi-agent approach allows for faster and more accurate vulnerability detection compared to traditional methods. Moreover, this approach provides high flexibility and adaptability to new cyber threats.

Overall, the study confirms that the use of multi-agent systems can significantly improve the effectiveness of penetration testing for web applications. Future research may focus on expanding the capabilities of such systems by integrating new agents for detecting other types of vulnerabilities and using machine learning techniques to adapt the agents to new threats.

Keywords: *multi-agent systems, penetration testing, XSS vulnerabilities, automated detection, web security.*

REFERENCES

[1] Kravari K. and Bassiliades N. A survey of agent platforms. J. Artif. Soc. Soc. Simul., vol. 18, 2015. DOI: 10.18564/jasss.2661.

[2] Gilbert N. and Bankes S. Platforms and methods for agent-based modeling. Proc. Natl. Acad. Sci. U. S. A., vol. 99, pp. 7197–7198, May 2002. DOI: 10.1073/PNAS.072079499.

[3] Railsback S., Lytinen S., and Jackson S. Agent-based simulation platforms: Review and development recommendations. Simulation, vol. 82, pp. 609–623, Sep. 2006. DOI: 10.1177/0037549706073695.

[4] Pal C.-V., Leon F., Paprzycki M., and Ganzha M. A review of platforms for the development of agent systems. Inf., vol. 14, p. 348, Jul. 2020. DOI: 10.3390/info14060348.

[5] Maneva R. Development of agent platform architecture for intelligent analysis of business processes. Bionics of Intelligence, 2020. DOI: 10.30837/bi.2020.1(94).09.

[6] Bragin A. Modern software tools for agent-based modeling. Artificial Societies, 2022. DOI: 10.18254/s207751800023501-0.

[7] Vrba P. JAVA-based agent platform evaluation. Sep. 2003, pp. 47–58. DOI: 10.1007/978-3-540-45185-3_5.

[8] Puliafito A., Tomarchio O., and Vita L. MAP: Design and implementation of a mobile agents' platform. J. Syst. Archit., vol. 46, pp. 145–162, Jan. 2000. DOI: 10.1016/S1383-7621(98)00076-9.

[9] Cabral L. Towards a theory of platform dynamics. ERN: Other Organizations & Markets: Personnel Management (Topic), Jul. 2018. DOI: 10.1111/jems.12312.

[10] Altmann D. and Gruber A. Using mobile agents in real world: A survey and comparison of different approaches. Semantic Scholar. [Online]. URL: https://surl.gd/tijhjn.

[11] Pan K., Lyu Y., and Pan Q. Adaptive formation for multiagent systems subject to denial-of-service attacks. IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, pp. 3391–3401, Aug. 2022. DOI: 10.1109/TCSI.2022. 3168163.

[12] Catta D., Leneutre J., Malvone V., and Murano A. Obstruction alternating-time temporal logic: A strategic logic to reason about dynamic models. pp. 271–280, 2024. DOI: 10.5555/3635637.3662875.

[13] Cavalcante R., Bittencourt I., Silva A., Silva M., Costa E., and Santos R. A survey of security in multi-agent systems. Expert Systems with Applications, vol. 39, pp. 4835–4846, Apr. 2012. DOI: 10.1016/j.eswa.2011.09.130.

[14] Yang X., Yang L., Dong L., Jin W.-H., Zhang M., Yang F., and Lin Y. Consensus tracking control for uncertain non-strict feedback multi-agent system under cyber-attack via resilient neuroadaptive approach. International Journal of Robust and Nonlinear Control, vol. 32, pp. 4251–4280, Feb. 2022. DOI: 10.1002/rnc.6035.

[15] Dorri A., Kanhere S., and Jurdak R. Multi-agent systems: A survey. IEEE Access, vol. 6, pp. 28573–28593, Apr. 2018. DOI: 10.1109/ACCESS.2018.2831228.

[16] Ahmed M., Kazar O., and Harous S. Cyber-physical system model based on multi-agent system. IET Cyber-Physical Systems: Theory & Applications, Jun. 2024. DOI: 10.1049/cps2.12096.

[17] Kantamneni A., Brown L., Parker G., and Weaver W. Survey of multi-agent systems for microgrid control. Engineering Applications of Artificial Intelligence, vol. 45, pp. 192–203, Oct. 2015. DOI: 10.1016/ j.engappai.2015.07.005.

[18] Gonçalves E., Cortés M., Campos G., Lopes Y.S., Freire E., Silva V., Oliveira K., and De Oliveira M.A. MAS-ML 2.0: Supporting the modelling of multi-agent systems with different agent architectures. Journal of Systems and Software, vol. 108, pp. 77–109, Oct. 2015. DOI: 10.1016/j.jss.2015.06.008.

[19] Calvaresi D., Appoggetti K., Lustrissimini L., Marinoni M., Sernani P., Dragoni A., and Schumacher M. Multi-agent systems' negotiation protocols for cyber-physical systems: Results from a systematic literature review. Pp. 224–235, 2018. DOI: 10.5220/0006594802240235.

[20] Binyamin S. and Slama S. Multi-agent systems for resource allocation and scheduling in a smart grid. Sensors, vol. 22, Oct. 2022. DOI: 10.3390/s22218099.

[21] Valentina A., Vishwashri R., and Rajadurai S. Finding Vulnerability in Web Application by using Pentesting. Int. J. Multidiscip. Res., 2024. doi: 10.36948/ijfmr.2024.v06i04.24517.

[22] Olivares-Naya M., de Gracia J.C., and S'anchez-Maci'an A. Adding web pentesting functionality to PTHelper. ArXiv, vol. abs/2410.12422, 2024. [Online]. URL: https://api.semanticscholar.org/CorpusID:273375081.

ТЗ. №4(36)’2025 https://doi.org/10.32782/2786-9024/v3i4(36).324435

[23] De Lima L., Horstmann M., Neto D., Grégio A., Silva F., and Peres L. On the Challenges of Automated Testing of Web Vulnerabilities. In 2020 IEEE 29th Int. Conf. Enabling Technol. Infrastruct. Collaborative Enterprises (WETICE), 2020, pp. 203–206. DOI: 10.1109/WETICE49692.2020.00047.

[24] Wijaya Y. Web-Based Dashboard for Monitoring Penetration Testing Activities Based on OWASP Standards. J. Ilm. Tek. Elektro Komput. Inform., 2020. DOI: 10.26555/jiteki.v16i1.17019.

[25] Doupé A., Cova M., and Vigna G. Why Johnny Can’t Pentest: An Analysis of Black-Box Web Vulnerability Scanners. 2010, pp. 111–131. doi: 10.1007/978-3-642-14215-4_7.

[26] Vimala K. and Fugkeaw S. VAPE-BRIDGE: Bridging OpenVAS Results for Automating Metasploit Framework. In 2022 14th Int. Conf. Knowl. Smart Technol. (KST), 2022, pp. 69–74. DOI: 10.1109/KST53302.2022.9729085.

[27] Albahar M., Alansari D., and Jurcut A. An Empirical Comparison of Pen-Testing Tools for Detecting Web App Vulnerabilities. Electronics, 2022. DOI: 10.3390/electronics11192991.

[28] Addressing Web Application Security Issues and Vulnerabilities Assessment Pen Testing. Int. J. Recent Technol. Eng., 2020. DOI: 10.35940/ijrte.f8169.038620.

[29] Al-Ahmad A., Ata B., and Wahbeh A. Pen Testing for Web Applications. Int. J. Inf. Technol. Web Eng., vol. 7, pp. 1–13, 2012. DOI: 10.4018/jitwe.2012070101.

[30] Bertoglio D.D., Gil A., Acosta J., Godoy J., Lunardi R., and Zorzo A. Towards new challenges of modern Pentest. ArXiv, vol. abs/2311.12952, 2023. DOI: 10.48550/arXiv.2311.12952.

[31] Bots: Introduction. Telegram. [Онлайн]. URL: https://surl.gd/ffcfcx. Accessed: 20.02.2025.

[32] Discovery P. Katana. GitHub. [Онлайн]. URL: https://surl.gd/tsgljg. Accessed: 20.02.2025.

[33] K.P., Gxss. GitHub. [Онлайн]. URL: https://surl.gd/bnuefy. Accessed: 20.02.2025.

[34] H. W., Dalfox. GitHub. [Онлайн]. URL: https://surl.gd/plpdtk. Accessed: 20.02.2025.

[35] Content Security Policy. [Онлайн]. URL: https://surl.gd/gpihew. Accessed: 20.02.2025.

[36] Web application firewall. Wikipedia. [Онлайн]. URL: https://surl.gd/efjsff. Accessed: 20.02.2025.

[37] Burp Suite. PortSwigger. [Онлайн]. URL: https://surl.gd/kteluz. Accessed: 20.02.2025.

[38] SPADE-MAS. Read the Docs. [Онлайн]. URL: https://surl.gd/uguutq. Accessed: 20.02.2025.