ТИТУЛЬНИЙ ЛИСТ

ТЕМА:

ПОЛІПШЕННЯ ЗАХИСТУ ВЕБ-ЗАСТОСУНКІВ ВПРОВАДЖЕННЯМ МОДЕЛІ БІЛОГО СПИСОКУ ПРИ РОЗРОБЦІ

СОКОЛОВ РУСЛАН МИКОЛАЙОВИЧ

ФТІ, КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ, БЕЗПЕКА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ

ГРУПА ФБ-52

НАУКОВИЙ КЕРІВНИК:

ГРАЙВОРОНСЬКИЙ МИКОЛА ВЛАДЛЕНОВИЧ

ЗАВДАННЯ

КАЛЕНДАРНИЙ ПЛАН

РЕФЕРАТ

Робота обсягом 46 сторінок містить 9 рисунків, 9 таблиць та 13 літературних посилань, 1 додаток.

Метою даної кваліфікаційної роботи є проектування та створення прототипу методики розробки захищених веб-застосунків на основі моделі білого списку.

Об’єктом дослідження є сучасні методи розробки захищених безпечних веб-застосунків

Предметом дослідження є поліпшення захисту і контроль за впровадженням механізмів захисту веб-застосунків.

Наукова новизна полягає в тому, що вперше було застосовано модель білого списку як методику розробки захищених веб-застосунків.

Результати роботи представлені методикою розробки захищених веб-застосунків, яка була успішно застосована і демонструє особливості безпеки впровадження моделі білого списку під час розробки.

Результати роботи можуть бути використані як додатковий модуль при проектуванні повноцінних методик розробки захищених застосунків для удосконалення механізмів захисту і контролем за їх працездатністю.

РОЗРОБКА ЗАХИЩЕНИХ ВЕБ-ЗАСТОСУНКІВ, МЕТОДИКИ РОЗРОБКИ ВЕБ-ЗАСТОСУНКІВ, БІЛИЙ СПИСОК, МОДЕЛЬ РОЗРОБКИ, PYTHON, DJANGO, БЕЗПЕКА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ

ABSTRACT

The work contains 46 pages, 9 figures, 9 tables, 13 references, 1 attachment.

The aim of this qualification work is to design method concept for secure web application development based on whitelist model.

The subject of the study is improving security and controlling web-application security implementation.

Scientific novelty is the fact that for the first time whitelist model was used as methodic for secure web-application development.

The results of the work are presented by secure development methodic which was successfully used and demonstrate security features of whitelist model implementation during development stage.

The results of the work can be used as additional module during designing completed secure web apps development methodology for improving and controlling web-application security implementation.

SECURE WEB-APPLICATION DEVELOPMENT, WEB-APPLICATION DEVELOPMENT METHODOLOGY, WHITELIST, DEVELOPMENT MODEL, PYTHON, DJANGO, SECURITY OF INFORMATION AND COMMUNICATION SYSTEMS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Веб-каркас або веб-фреймворк (англ. Web application framework, WAF) – це каркас, призначений для створення динамічних веб-сайтів, мережевих додатків, сервісів або ресурсів. Він спрощує розробку і позбавляє від необхідності написання рутинного коду. Багато каркасів спрощують доступ до баз даних, розробку інтерфейсу, і також зменшують дублювання коду [1].

Model View Controller (MVC, «Модель-Представлення-Контролер», «Модель-Вид-Контролер») – схема поділу даних програми, призначеного для користувача інтерфейсу і керуючої логіки на три окремих компоненти: модель, уявлення і контролер - таким чином, що модифікація кожного компонента може здійснюватися незалежно [2].

API (англ. Application Programming Interface) – це набір визначень, протоколів та інструментів для розробки ПЗ і додатків. API-інтерфейс розробляється для спрощення створення програм, шляхом надання всіх необхідних функціональних блоків. API може бути розроблено для web систем, ОС і баз даних, надаючи середовище для розробки додатків використовуючи конкретну мову програмування.

БД (база даних, англ. Database) – сукупність даних, організованіх відповідно до концепції, яка опісує характеристику цих даних і взаємозв'язки між її елементами.[3]

Open Web Application Security Project (OWASP) – це відкритий проект забезпечення безпеки веб-додатків.

ВСТУП

Інформаційні технології бурно розвиваються і захоплюють усе більше сфер нашої життєдіяльності. Багато існуючих організацій базуються в основному лише IT сфері. Все більше даних різного роду чутливості стають доступними через інтернет мережу через це ми виграємо багато годин нашого життя які ми провели б у чергах у банку, за пошуком потрібної книги у бібліотеці.

Це все доступно завдяки веб-застосункам, мобільним додаткам які надають ті чи інші сервіси.

У новинах постійно з'являється інформація про успішні кібер-атаки. Тому під час розробки веб-застосунків на перший план виходить забезпечення його безпеки. Сьогодні багато веб-застосунків мають загальновідомі вразливості за допомогою яких можна успішно проводити атаки на веб-ресурси.[4]

Відповідно до досліджень 2016 року 97% JAVA стосунків мають у собі хоча б одну вразливість. Відповідно до інших досліджень, вразливості безпеки різного рівня критичності присутні що наймеше у 90% веб-застосунках і мобільних за стосунках. [5]

Сьогодні питання безпеки у веб-застосунках вирішується за допомогою фреймоврків за допомогою яких вони були створені, та деяких інших стороніх програм, різних фаєрволів, IPS, антивірусів які були розроблені як додаткові міри впровадження захисту веб-застосунків або як частина самого за стосунку.

Відповідальність за безпеку веб-застосунків несуть в першу чергу самі розробники. Зазвичай розробка веб-застосунків включає в себе команду розробників та деяку визначену методку розробку програмного забезпечення, але не всі методики розробки веб-застосунків виносять на перший план безпеку.

Актуальність даної роботи зумовлюється тим, що розробка веб-застосунків потребує впровадженю безпеки уже з перших кроків розробки застосунку.

Сучасні методики розробки програмного забезпечення, а особливо методики розробки веб-застосунків не завжди виносять безпеку на перший план з міркувань більшої затрати ресурсів і часу на її ефективне впровадження. Представлена робота пропонує легкий, гнучкий і ефективний метод забезпечення безпеки на основі моделі білого під час розробки веб-застосунків який можна адаптувати в будь-яку існуючу методику розробки, дана методика враховує аспекти впровадження захисту уже на перших кроках розробки веб-застосунків.

Метою роботи є поліпшення впровадження безпеки веб-застосунків на етапі розробки.

Завданням роботи є дослідження проблеми створення захищених веб-застосунків а також сучасних методів і прийомів розробки веб-застосунків, створення концепту методики на основі моделі білого списку яка буде рішеням розробки безпечних веб-застосунків, застосування даної моделі під час розробки та перевірка працездатності механізмів захисту які забезпечує розробка по даній методиці.

Об’єктом дослідження є сучасні засоби і методи розробки безпечних веб-застосунків та можливі шляхи поліпшення існуючих.

Наукова новизна визначається тим, що в результаті роботи було запропоноване не типове рішеня для процесу розробки безпечних веб-застосунків на основі концепції білого списку.

1 РОЗРОБКА БЕЗПЕЧНИХ ВЕБ ЗАСТОСУНКІВ

У даному розділі буде проаналізована проблема створення безпечних веб-заастосунків, деякі рекомендації від OWASP що до розробки безпечних веб-застосунків проаналізовані типові та найчастіші загрози і вразливості веб-застосунків.

1.1 Аналіз проблеми безпечного процесу розробки веб-застосунків

Створення безпечних веб-застосунків задача еквівалентна створенню безпечного програмного забезпечення.

Створення безпечних веб-застосунків – комплексна задача в якій можна виділити проблеми програмної інженерії, менеджменту і інженерії комп'ютерної безпеки. Для розробки безпечного веб-застосунку необхідні вміння з програмної інженерії, а також знання технічної частини.

Декілька прикладів впровадження безпеки під час процесу розробки.

Багато процесів розробки які використовуються найчастіше були адаптовані і змінені для забезпечення безпеки. Бо створення нових методів розробки з нуля потребують багато коштів і часу. Тому розробники часто змінюють або доповнюють уже існуючі для вирішення питань безпеки. [6]

Наприклад поєднання концепції розробки від Microsoft SDL «Життєвий цикл безпечної розробки» з методикою «Гнучкої розробки». Який використовує каскадну структуру SDL і розбиває вимоги на спрінти, вимоги які повинні виконуватися на протязі усього життєвого циклу і на ті шо виконуються один раз. Або як інший приклад використовувати Microsoft SDL у сукупності з екстремальною методикою розробки програмного забезпечення.

Альтернативою такому поєднанню для безпечної розробки програмного забезпечення є адаптація методів безпеки у концепцію розробки. Таким чином команда розробників пристосовує свій процес розробки, щоб задовільнити безпеку як не функціональну вимогу.

Деякі складові розробки безпечного програмного забезпечення є життєво важливими і розробляються на ранніх етапах розробки. Наприклад створення моделі загроз яка буде враховуватись при визначені наступних етапів розробки, такий підхід часто застосовується в каскадних методиках розробки програмного забезпечення. Такий підхід не завжди використовується у гнучкій методиці розробки. У гнучкій розробці модель загроз сприймається як існуючі артефакти які виправляються під наступних ітерацій або спринтів Найпопулярніший спосіб впровадження безпеки до гнучкої розробки використання так званних безпечних спринів (security sprints).

1.2 OWASP рекомендації щодо розробки безпечного коду

Open Web Application Security Project (OWASP) – це відкритий проект забезпечення безпеки веб-застосунків. Товариство OWASP включає в себе корпорації і наукові заклади багатьох країн. OWASP працює над створенням статей, навчальних посібників, рекомендацій, документацій, інструментів і технологій, які зберігаються у відкритому доступі.

OWASP рекомендації щодо безпечної розробки коду описує деякі прийоми і методи які розробники можуть використовувати для створення захищених за стосунків які забезпечують конфіденційність, цілісність і доступність інформації[7].

- Валідація вхідних даних.

- Кодування вихідних даних відносно належного контексту.
- Автентифікація і менеджмент паролів.
- Менеджмент сеансів.
- Конроль доступу.
- Криптографічний захист.
- Обробка помилок і легування.
- Захист чутливих даних.
- Безпека зв’язку.
- Системна конфігурація.
- Безпека бази даних.
- Менеджмент файлів.
- Керування пам’яттю.
- Загальні прийоми кодування.

1.3 Аналіз загроз безпеки веб-застосунікв

Безпека веб-застосунікв це розділ інформаційної безпеки який відповідає за вирішення питань безпеки веб-сайтів, веб-застосунків та веб-сервісів. На високому рівні безпека веб-застосунків базується на принципах безпеки прикладних програм які мають доступ до інтернету.

Більшість атак на веб-застосунки реалізуються шляхом міжсайтового скриптингу (XSS) і SQL-ін’єкцій які зазвичай можливі через недостатню професійність кодера і помилками зв’язаними з обробкою застосунком вхідних і вихідних даних. Зазвичай ці дві загрози стоять на початку усіх чартів які надають інформацію про найчастіші загрози які є наслідком погано написаного коду.

Відповідно до постачальника послуг безпеки Cenzic, топ загроз за березень 2012 року має такий вигляд

Таблиця 1.1. – Топ загроз веб-застосунків від Cenzic за березень 2012[8]

Відносна частота загрози	**Назва загрози**
37%	Міжсайтовий скриптинг
16%	SQL ін'єкції
5%	Розкриття повного шляху у get запитах
5%	Data breach (information disclosure)
4%	Виконання довільного коду
4%	Пошкодження пам'яті
4%	Cross-site request forgery
3%	Розкриття конфіденційної інформації
3%	Довільне виконання файлів
2%	Виконання локальних файлів серверу
1%	Віддалене виконання файлів
1%	Переповнення буферу
15%	Інші, включаючи ін'єкції JS-коду

1.4 OWASP TOP 10

Один з найбільших вкладів OWASP вніс за рахунок проекту Top Ten Vulnerabilities – список найбільш частих 10 загроз веб-застосунків, зазвичай список оновлюється з періодом 3 роки[9].

1. Ін'єкції – Injections

Додатки використовують SQL-запити для того, щоб отримувати, додавати, змінювати або видаляти дані, наприклад при редагуванні користувачем своїх особистих даних або заповненні анкети на сайті. При недостатній перевірці даних від користувача, зловмисник може впровадити в форму Web-інтерфейсу додатку спеціальний код, що містить шматок SQL-запиту.

2. Недоліки системи аутентифікації і зберігання сесій (Broken Authentication and Session Management)

У разі, якщо ваш ідентифікатор вкраде зловмисник, а в системі не були реалізовані перевірки, скажімо IP-адреси сесії, або перевірки наявності більш одного з'єднання в одній сесії, зловмисник зможе отримати доступ до системи з правами вашого облікового запису. А якщо це інтернет-банк або кабінет платіжної системи, про наслідки такого несанкціонованого доступу Ви можете легко здогадатися самі.

3. Міжсайтовий скриптинг - XSS (Cross Site Scripting)

Міжсайтовий скриптинг - ще одна помилка валідації призначених для користувача даних, яка дозволяє передати JavaScript код на виконання в браузер користувача. Атаки такого роду часто також називають HTML-ін'єкціями, адже механізм їх впровадження дуже схожий з SQL-ін'єкціями, але на відміну від останніх, впроваджуваний код виповнюється в браузері користувача.

4. Небезпечні прямі посилання на об'єкти (Insecure Direct Object References)

Даний вид уразливості є також наслідком недостатньої перевірки призначених для користувача даних. Суть її полягає в тому, що при виведенні будь-яких конфіденційних даних, наприклад особистих повідомлень або облікових карток клієнтів, для доступу до об'єкта використовується ідентифікатор, який передається у відкритому вигляді в адресному рядку браузера, І не реалізована перевірка прав доступу до об'єктів.

5. Небезпечна конфігурація (Security Misconfiguration)

Безпека Web-додатки вимагає наявності безпечної конфігурації всіх компонентів інфраструктури: компонентів програми (таких як фреймворки - frameworks), веб-сервера, сервера баз даних і самої платформи. Налаштування компонентів сервера за замовчуванням найчастіше небезпечні і відкривають можливості до атак. Наприклад, крадіжка сесійного cookie через JavaScript при XSS-атаки стає можливою завдяки виключеною за замовчуванням налаштування cookie_http only.

6. Незахищеність критичних даних (Sensitive Data Exposure)

Багато веб-додатки не захищають конфіденційні дані, такі як кредитні карти і облікові дані для аутентифікації. Зловмисники можуть вкрасти або модифікувати такі слабо захищені дані для використання в своїх корисливих цілях.

7. Відсутність рівнів функцій контролю доступу (Missing Function Level Access Control)

Суть уразливості, як випливає з назви, полягає у відсутності перевірки наявності належного доступу до запитуваного об'єкту.

8. Міжсайтова підробка запиту (Cross-Site Request Forgery, CSRF / XSRF)

Вектор атаки CSRF, також відомий як XSRF, дозволяє зловмиснику виконувати від імені жертви дії на сервері, де не реалізовані додаткові перевірки.

9. Використання компонентів з відомими уразливими (Using Components with Known Vulnerabilities)

Найчастіше web-додатки написані з використанням спеціальних бібліотек або «фреймворків» (англ - framework), які поставляються сторонніми компаніями. У більшості випадків ці компоненти мають відкритий вихідний код, а це означає, що вони є не тільки у вас, але і у мільйонів людей у всьому світі, які студіюють їх вихідний код, в тому числі, і на предмет вразливостей. І потрібно відзначити, що роблять вони це аж ніяк не безуспішно.

10. Невалідовані переадресації та пересилання (Unvalidated Redirects and Forwards)

Web-додатки часто переадресовують користувача з однієї сторінки на іншу. В процесі можуть використовуватися неналежним чином перевіряються параметри із зазначенням сторінки кінцевого призначення переадресації.

OWASP Топ десять вразливостей	Django	Rails	CakePHP
A1: Ін'єкції			
A2: Недоліки системи аутентифікації і зберігання сесій			
A3: Міжсайтовий скриптинг			
A4: Небезпечні прямі посилання на об'єкти			
A5: Небезпечна конфігурація			
A6: Незахищеність критичних даних			
A7: Відсутність рівнів контролю доступу функцій			
A8: міжсайтова підробка запиту			
A9: Використання компонентів з відомими уразливими			
A10:Невалідовані переадресації та пересилання			

Рисунок 1.1 – вбудовані механізми захису від загроз OWASP TOP деяких фреймворків

Висновок розділу 1

У даному розділі буде проаналізовано проблема створення безпечних веб-заастосунків, деякі нюанси їх створення а також типові загрози.

Проаналізувавши проблему створення веб-застосунків можна зробити висновок що ця проблема актуальна і обширна і має спільні риси з загальною концепцією створення безпечного прикладного забезпечення. Дана проблема частково залежить від механізмів захистувикористовуваних веб-каркасів на яких будується веб-застосунок, використовуваній базі даних, безпеки в цілому від сервера на якому виконується веб-застосунок і найбільше від професійності розробника.

2 КОНЦЕПЦІЯ БІЛОГО СПИСКУ

У даному розділі буде розглянена загальна концепція білого списку, типове використання білого списку у ІТ і також створена методика розробки безпечних веб-застосунків на основі математично описаної моделі білого списку, а також простий приклад використання даної методики.

2.1 Типове використання методу білий список для забезпечення безпеки у веб-застосунках

Концепція білого списку загально відома і використовується досить давно у багатьох сферах ще до появи інформаційних і цифрових технологій.

Білий список можна визначити так:

«Білий список» — практика ідентифікації суб'єктів, яким надаються певні привілеї, послуги, мобільність, доступ або визнання. Суб'єкти в списку будуть прийняті, схвалені та /або будуть визнані. «Білий список» — це протилежність чорних списків, практика ідентифікації осіб, які заперечуються, не визнаються.

Часто модель білого списку використовують для захисту від спаму або шкідливого програмного забезпечення, для налаштування доступу до мережі або під час валідації вхідних і вихідних данних, об’єктів будь-якого роду.

Наприклад, ми хочемо налаштувати віддалений доступ до бази даних яка знаходиться у хмарі. За для підвищення рівня безпеки ми в налаштуваннях віддалених підключеннях додамо нашу IP адресу з якої хочемо підключитися до бази даних у білий список, такий простий механізм, виключить можливість підключиться до нашої хмари з усіх комп’ютерів окрім того, де IP адреса знаходиться у білому списку.

Білий список можна визначити як деяка множина елементів (наприклад, програми, список електронних адрес, або деякі дані) яким надається право чи спеціальна привілегія на виконання або на обробоку чи доступ до чогось.

Протилежним до такого методу є метод Чорного списку – множина деяких елементів у яких відібрані відповідні права.

Білий список як і чорний список активно застосовуються у різноманітних механізмах захисту мережевих інфраструктурах та системах.

2.2 Ідея використання білого списку під час безпечної розробки

Ціль данної роботи використання моделі білого списку для забезпечення безпечної розробки веб за стосунків. Ідея полягає в наступному: створити деяку модель білого списку яка буде буде визначати усі дозволені операції веб-застосунку розробленого за допомогою каркасу на основі Model View Controler (Модель Представленя Контролер). Усі операції які не були включені до білого списку будуть повертати виконання операції у безпечний стан.

В теорії використання моделі білого списку під час безпечної розробки веб-застосунків дозволяє запобігати деяким вразливостям які ігноруються усіма мені відомими веб-каркасами а за наявність яких відповідальні саме розробники, відповідно до моїх досліджень. Як кінцевий результат мого дослідження я спробую максимально об’єктивно оцінити даний підхід.

Головна мета показати що створений застосунок відповідає концепції білого списку і використаний метод запобігає деяким вразливостям а саме A4 відсутність рівнів функцій контролю доступу і A7 небезпечні прямі посилання на об'єкти з OWASP TOP 10.

2.3 Опис моделі білого списку розробки безпечних веб-застосунків

Веб-фреймворки запобігають більшості відомих вразливостям веб-додатків таких як SQL- ін’єкції, XXS, але вони не можуть запобігти деяким специфічним вразливостям які притаманні саме конкретній програмі яка розробляється деяким фреймворком, відповідальність за запобігання таких вразливостей залишається за розробниками. Так як постійно хакери знаходять все більш вишукані методи експлойту вразливостей, розробникам потрібно вкладувати все більше ресурсів на впровадження безпеки.

В теорії, методика білого списку розробки веб-застосунків дозволить розробникам підвищити рівень безпеки шляхом виявлення непередбачуваної і небезпечної поведінки веб-застосунку, припиненням її або перенаправляючи на безпечну.

Невідповідність між очікуваною і реальною поведінкою веб-застосунку може являтися індикатором атаки на веб-застосунок. Дана робота зосереджена на OWASP 4 і 7 вразливостях: небезпечні прямі посилання на об'єкти. відсутність контролю доступу функцій. Мета моєї роботи реалізувати механізм безпеки для веб-застосунку шляхом передбачення дозволених операцій. Даний підхід був застосований частково абстрактно. Якщо розробляти веб-застосунок шляхом забезпечення реалізації тільки дозволених взаємодій між компонентами за стосунку, то це значно спростить моніторинг безпеки. Будь-яка послідовність дій застосунку яка не була в специфікації розробки буде відхилена логікою застосунку і перенапрямлена.

Дана робота визначає, створює і впроваджує список дозволених взаємодій веб-застосунку. Ці правила будуть керувати HTTP запитами і відповідями які обробляє веб-застосунок. Визначення елементів які входять до білого списку повинно розроблятися під час етапу проектування.

В ідеальній концепції, білий список повинен створюватися динамічно під час розробки веб-застосунку шляхом використання інструментів моніторингу поведінки і виконання програми. Але оскільки наша головна ціль вияснити чи буде покращена безпека веб-застосунку, використаємо статичний білий список який визначемо на етапі проектування. Реалізація білого списку можлива при створені веб-застосунку використовуючи будь-який фреймворк на основі MVC так як одна з особливостей роботи таких фреймворків має у собі механізм який працює за концепцією посередника і відповідає за перехоплення HTTP запиту і видачу дозволеної відповіді, яка задається відповідною моделлю білого списку.

2.4 Формальне визначення моделі білого списку розробки безпечних веб-застосунків

Визначимо білий список як набір чотирьох множин {C, D, W, S}, де:

C множина елементів $\{u, c_1, c_2,\ldots,c_n\}$, де $c_1, c_2,\ldots,c_n$ – складові компонент які входять в межі системи, а u компоненти які за межею системи.

D множина елементів $\{d_1, d_2,\ldots,c_n\}$, де $d_1, d_2,\ldots,c_n$ стани компонентів.

Кожна комірка матриці розмірності |C| X |C| містить унікальну підмножину x, $\{x: x \subseteq D\}$. Якщо оцінка станів x повертає 1, то упорядкована пара переходу стану з $c_{\text{початковий(origin)}}$ до $c_{\text{кінцевий(destination)}}$ додається до множини W.

W множина впорядкованих пар $\{(c_o, c_d) : c_o, c_d \in C\}$ кожна пара представляє собою перехід з початкового компоненту c_0 до кінцевого c_d

S матриця розмірності |C| X |C| $S_{co,\ cd} = c_{\text{безпечний(safe)}}$ визначає безпечні компоненти $\{c_s : c_s \in C\}$ де c_d не може слідувати з c_o.

Перехід з однієї складової до іншої керується функцією переходу де перехід з $c_{\text{початковий}}$ до $c_{\text{кінцевий}}$ відбувається тоді і тільки тоді якщо $(c_o, c_d) \in W$, інакше функція переходу визивається через $(c_o, S_{co, cd})$.

$$T(c_0, c_d) = \begin{cases} c_d, \text{якщо } (c_0, c_d) \in W \text{ інакше,} \\ T(c_0, S_{c_0 c_d}) \end{cases}$$

Визначено кілька операцій які виконуються за допомогою білого списку. Операції поділені на дві категорії відповідно до того коли вони можуть бути застосовані. Наступні операції будуть використовуватися під час розробки:

- Створення (c_o, c_d) у множині W: додавання впорядкованої пари до множини
- Видалення (c_o, c_d) з множини W: видалення впорядкованої пари з множини
- Ввід {d_x} до множини D: додавання станів d_x до множини D.
- Видалення {d_x} з множини D: видалення станів d_x з множини D.
- Додавання {d_x} до підмножини x множини $W_{co,cd}$.
- Видалення {d_x} з підмножини x множини $W_{co,cd}$.
- Ввід {c_s} у комірку матриці $S_{co,cd.}$
- Оновлення {c_s} у комірці матриці.

Операції білого списку які дозволені при виконані:

- Вичислення T(c_o, c_d)
- Варифікація $c_o \rightarrow c_d$:

c_d може слідувати до c_o якщо дане твердження хибне, то усі стани які належать підмножині x у $W_{co,cd}$ будуть повертати істину.

Інакше перехід до безпечного стану c_s.

Множина усіх компонентів C і відношень W можуть бути представлені у вигляді двійкової матриці, де 1 означає дозволений перехід стану а 0 означає не дозволений. Кожна комірка матриці прийматиме значення або 0 або 1 відповідно до значень підмножини станів. Матриця S міститиме безпечні переходи станів у випадку коли перехід стану з c_o до c_d не буде дозволений.

2.5 Кроки впровадження моделі

В першу чергу потрібно ідентифікувати дозволену поведінку веб-застосунку шляхом створення діаграми яка буде відображати яким чином поводить себе програма. Діаграма повина відображати усі дозволені взаємодії між компонентами застосунку. Потрібно дослідити кожну операцію і ідентифікувати підмножину переходів станів (c_o,c_d) і помістити їх у відповідні комірки матриці W. Також потрібно ідентифікувати безпечні компоненти і переходи у разі повертанні хибного значення перевірки дозволеності переходу стану. Безпечні компоненти c_s слід помістити у комірки які відповідають (c_o,c_d) у матриці W до матриці S. Одже на даному етапі розробки ми маємо підмножини переходу станів які записані у матрицю W і приймають значення 1 або 0 в залежності від дозволеності переходів станів. Для простоти назвемо таке представлення як M, де |C| X |C| = M де $M_{co,cd}$ = 1 якщо (c_o,c_d) є W і $M_{co,cd}$ = 0 якщо (c_o,c_d) не належить W. Модель білого списку може змінюватися або доповнюватися в залежності від ходу розробки. Варто також зазначти що, стани в множині D повинні бути простими а не комплексними.

Давайте визначимо наступні кроки побудови моделі білого списку:

- Створити діаграму яка відображає назначену допустиму поведінку веб-застосунку
- Задати підмножини станів за для змоги аналізу і назначення дозволених переходів і взаємозв’язків застосунку
- Розмістити кожну підмножину переходу станів c_o,c_d відповідну комірку матриці |C|x|C|
- Ідентифікувати безпечні компоненти c_s і розмістити їх у комірки які відповідають (c_o,c_d) у матриці W до матриці S.
- Присвоїти значення 1 або 0 для кожної підмножини переходу станів в залежності відповідності білого списку:

- $M_{co,cd}$ = 1 якщо (c_o,c_d) є W і $M_{co,cd}$ = 0 якщо (c_o,c_d) не належить
- Належним чинном налаштувати процес розробки для впровадження методики білого списку в уже існуючий процес розробки на будь-якому етапі.

2.6 Приклад застосування

Для наглядної роботи застосування концепту білого списку при розробці веб-за стосунків використаємо наступний банальний приклад. Допустимо у нас є веб-застосунок який вимагає обов'язкову аутентифікацію користувача для можливості його використання. Нехай застосунок дозволяє 3 спроби аутентифікації. Якщо користувач провалить 3 спроби, застосунок заблокує доступ. Якщо користувач успішно пройде аутентифікацію застосунок пере направить користувача на його персональну сторінку. Користувач матиме змогу редагувати свій профіль або зв'язатися з іншими користувачами. Користувач також має змогу вийти зі свого профілю в будь-який час.

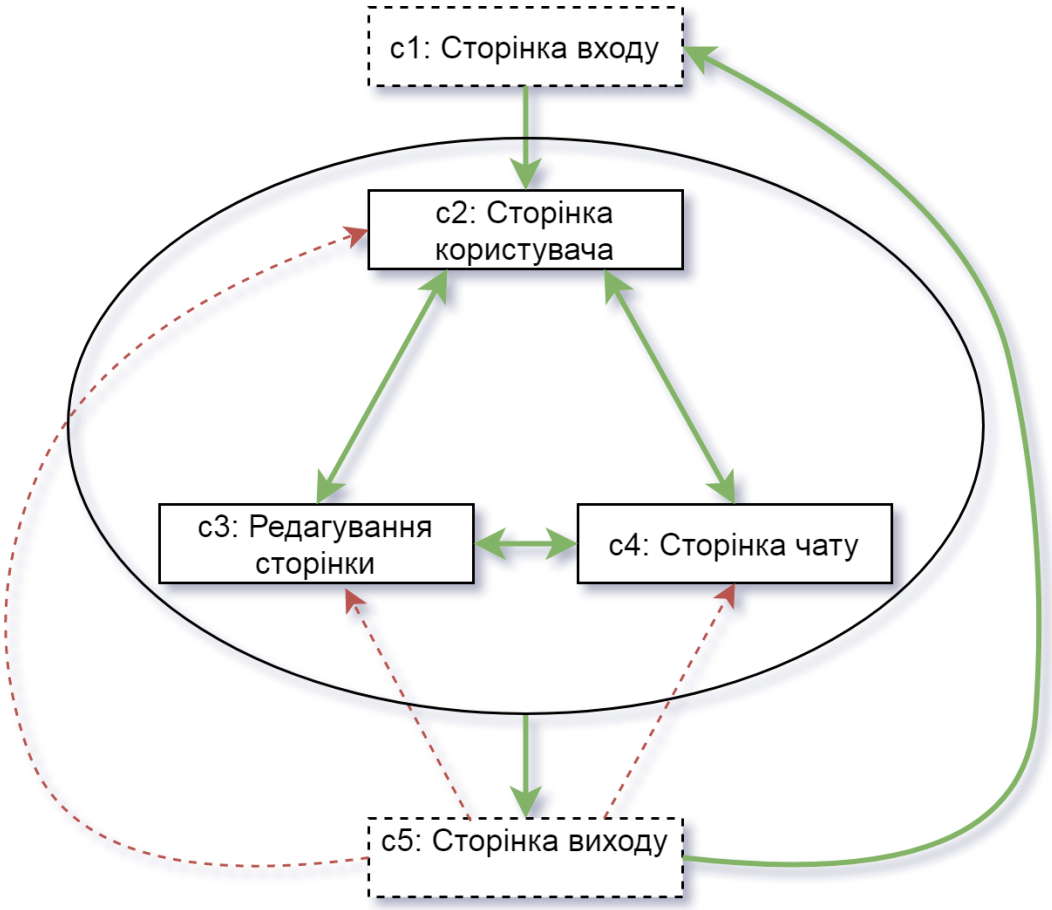

Рисунок 2.1 – Приклад діаграми яка відображає поведінку програми

Зелені стрілки дозволені операції білим списком. Червоні не дозволені

Модель за стосунку даного прикладу складається з 5 компонент. C = {u, c_1, c_2, c_3, c_4, c_5}. U представляє собою компоненти за границею системи і включається до C за для повноти. Щодо глобальної множини станів D, припустимо що вони означають наступні стани:

- d1: користувач анонімний.
- d2: коистувач авторизований.
- d3: час існування сесії валіний.
- d4: попереднє представлення.
- d5: представлення підпослідовності
- d6: спроби аутентифікації < 3.

Білий список містить підмножину D у кожній комірці матриці. Для прикладу, білій список нижче відображає підмножину дозволених переходів станів з c_1, c_2 і іншу підмножину яка приводить до недозволеного переходу з c_5 до c_4. Для дозволеного переходу з c_1 до c_2, підмножина станів: x = {d_2, d_3, d_4 = login view, d_5 = user portal view, d_6}. Усі стани у x мають повертати істину. Для недозволених переходів з c_5 до c_4, підмножина станів x = {d_2, d_3, d_4 = edit profile view або user portal view, d_6}.

Очевидно, перехід з стану c_5 до c_4 не дозволений, бо перший стан у підмножині d_2 не може бути досягненим якщо користувач вийшов з аканту. Заповнимо наші переходи у матрицю переходів станів.

Таблиця 2.1 – матриця W впорядкованих пар станів переходу компонентів

	u	**c1**	**c1**	**c3**	**c4**	**c5**
u	-	-	-	-	-	-
c1	-	-	x = {d_2, d_3, d_4 = login view, d_5 = user portal view, d_6}	-	-	-
c2	-	-	-	-	-	-
c3	-	-	-	-	-	-
c4	-	-	-	-	x = {d_2, d_3, d_4 = edit profile view або user portal view, d_6}	-
c5	-	-	-	-	-	-

Нехай множина впорядкованихпар переходу станів наступна

W = {(u; u) ; (u; c1) ; (c1; c1) ; (c1; c2) ; (c2; c2) ; (c2; c3) ; (c2; c4) ; (c2; c5) ;

(c3; c2) ; (c3; c3) ; (c3; c4) ; (c3; c5) ; (c4; c2) ; (c4; c3) ; (c4; c4) ; (c4; c5) ; (c5; c1) }

Тоді можна подати представлення матриці W у двійкому вигляді:

Таблиця 2.2 – двійкова матриця переходу станів компонентів W

	u	**c1**	**c2**	**c3**	**c4**	**c5**
u	1*	1	0	0	0	0
c1	0	1	1	0	0	0
c2	0	0	1	1	1	1
c3	0	0	1	1	1	1
c4	0	0	1	1	1	1
c5	0	1	0	0	0	0

Наступник крок це запонити матрицю S безпечними компонентами для пере направлення переходу на безпечний стан якщо він не дозволений.

Таблиця 2.3 – матриця переходу веб-сторінок у безпечні стани

	u	**c1**	**c2**	**c3**	**c4**	**c5**
u	NA	c1	c1	c1	c1	c1
c1	c5	c1	c1	c1	c1	c1
c2	c5	c5	c5	c1	c1	c5
c3	c5	c5	c5	c5	c5	c5
c4	c5	c5	c5	c5	c5	c5
c5	c5	c5	c5	c5	c5	c5

Висновок розділу 2

У даному розділі було проаналізована концепія типового використання білого списку у ІТ а також розроблена модель білого списку і методика використання даної моделі для розробки безпечних веб-застосунків. А також приклад використання. Можна зробити висновок, що будь-яку концепцію захисту чи розмежування доступу можна адаптувати під будь-який процес, а саме під процес розробки захищених веб-застосунків. Згідно аналізу створеного прикладу модель являється працюючою і успішно забезпечує безпеку зпроектованого за стосунку.

3 РОЗРОБКА І ПЕРЕВІРКА ВЕБ-ЗАСТОСУНКУ З ВИКОРИСТАННЯМ МОДЕЛІ БІЛОГО СПИСКУ

У даному розділі буде розроблений веб-застосунок за допомогою створеного в даній роботі методу розробки захищених веб-застосунків білий список і також перевірка веб-застосунку на відповідність до моделі білого списку. Для застосування моделі білого списку створимо веб-програму за допомогою Python веб-фреймворку Django. При створені програми були використані найновіші версії програм, а саме Python версії 3.7.2 і Django 2.1.7. Для простоти реалізації створимо веб-застосунок у вигляді форума. Для перевірки веб-застосунку на відповідність до моделі білого списку використовувались інструменти OWASP ZAP.

3.1 Обгрунтування вибору веб-фреймворку і його можливості

Django (Джанго) - фреймворк у вільному доступі для створення веб-додатків на мові Python, що використовує шаблон проектування MVC. Проект підтримується організатором Django Software Foundation.

Сайт на Django будується з одного або декількох програм, які рекомендується робити роздільними. Це одна з істотних архітектурних відмінностей цього фреймворка від деяких інших (наприклад, Ruby on Rails). Один з основних принципів фреймворка - DRY (англ. Don’t repeat yourself)

Також, на відміну від інших фреймворків, обробники URL в Django конфігуруются явно за допомогою регулярних виразів.

Для роботи з базою даних Django використовує власний ORM, в якому модель даних описується класами Python, і по ній генерується схема бази даних. [10]

Деякі можливості Django:

- ORM, API доступу до БД з підтримкою транзакцій
- вбудований інтерфейс адміністратора, з уже наявними перекладами багатьма мовами
- диспетчер URL на основі регулярних виразів
- розширювана система шаблонів з тегами і спадкуванням
- система кешування
- інтернаціоналізація
- підключається архітектура додатків, які можна встановлювати на будь-які Django-сайти
- «Generic views» - шаблони функцій контролерів
- авторизація та аутентифікація, підключення зовнішніх модулів аутентифікації: LDAP, OpenID та ін.
- система фільтрів («middleware») для побудови додаткових обробників запитів, як наприклад включені в дистрибутив фільтри для кешування, стиснення, нормалізації URL і підтримки анонімних сесій
- бібліотека для роботи з формами (успадкування, побудова форм по існуючої моделі БД)
- вбудована автоматична документація по тегам шаблонів і моделей даних, доступна через панель адміністрування

Django каркас веб-застосунків створений на мові Python. Веб-фреймворк це набір програм за допомогою якого можна створювати динамічні веб-сайти, за стосунки і сервіси. Веб-каркас надає набір інструментів і функціонал за допомогою яких можна вирішити багато найпоширеніших проблем пов’язаних з вебр-розробкою, а саме конфігурація і забезпечення безпеки, проблеми зв’язані з базами даних, менеджментом сесій, шаблонами веб-сторінок, URL маршрутизація, локалізація та інші.

Використовуючи веб-фреймворк, такий як Django дозволяє нам розробляти безпечні і надійні веб-застосунки відносно досить швидко і по всім стандартам та правилам, при цьому не витрачаючи час на написання загально використовуваних частин програм

Django однозначно включає увесь необхідний функціонал і пропонує широкий спектр функціоналу, наприклад, вбудований веб-сервер для розробки і тестування, з'єднуюче програмне забезпечення (middleware system), можливість кешування, інструментарій для роботи з базами даних(ORM), можливість використання шаблонів (template engine), оброку форм, інструменти для Unit-тестування, безпечна вбудована система налаштування аутентифіцації, панель адміністрування, вбудована підтримка RSS і GIS.

Ще одна з особливостей в тому що Django веб-фреймворк мови Python. А значить для нього існує величезна кількість бібліотек у відкритому доступі. Тому для рішення типових задач можна використовувати фунціонал встановлених бібліотек.

3.2 Основні механізми безпеки DJANGO

Фреймворк Django один з найпопулярніших фреймворків, одна з причин це надійні і легко реалізовувані механізми впровадження безпеки веб-застосунків.

Таким чином Django впроваджує захист від міжсайтового скриптингу:

Використання шаблонів Django дозволяє захищати за стосунок від більшості атак типу міжсайтового скриптингу. Але потрібно розуміти яким чином впроваджується захист і які обмеження при цьому накладаються[11].

Існує шаблон Django для виключення спеціальних символів в формах, які дозволено використовувати в HTML. Цей шаблон захищає від більшості відправлених потенційно шкідливих форм, але запис такого виду:

<style class={{ var }}>...</style>

Все ж дозволений. Якщо змінна var відноситься до 'class1 onmouseover=javascript:func()', то це приведе до неавторизованого виконання JavaScrip.

Також важливо бути обережним при використанні is_safe у користувацьких шаблонних тегаг safe і mark_safe і коли авто пропуск спеціальних символів виключений.

Для захисту від міжсайтової підробки запиту (Cross site request forgery) має деяку програму посередник – CsrfViewMiddleware:

Django перевіряє деякий серкет у кожному POST запиті, так званний csrf token, який зберігається у cookie. Якщо сайт використовує HTTPS, то CsrfViewMiddleware перевірить чи HTTP заголовок встановлений у URL має однаковий піддомен і порт.

Захист об’єктів бази даних від SQL-ін’єкцій у Django реалізований за допомогою параметризації самих об’єктів бази даних. Тобто SQL код створюється окремо від параметризації об’єктів бази даних. Так як параметри можуть бути поставлені користувачем, а звідси вони можуть бути потенційно небезпечними, то вони пропускаються механізмом керування бази даних.

Також Django має програму-посередник X-Frame-Options middleware яка запобігає вразливостям типу клікджекінгу(англ. – clickjacking). Запобігає рендерінгу стороннього сайту у рамці на поточній сторінці. Цей механізм можна вимкнути або налаштувати на дозволені рендери сайтів на поточній сторінці.

Завжди краще розвертати сайт з використанням HTTPS. В інакшому випадку існує можливість для користувачів перехопити дані аутентифікації, або будь-яку іншу інформацію яка передажться з сервера на клієнт, або навіть в деяких випадках існує можливість змінювати дані які передаються по обох напрямках. Для впровадження механізмів захисту які надаються разом з HTTPS потрібно виконати наступні кроки:

- Встановити для SECURE_PROXY_SSL_HEADER значення True.
- Встановити SECURE_SSL_REDIRECT значення True, таким чином запити HTTP будуть перенаправлятись до HTTPS.
- Встановити SESSION_COOKIE_SECURE і CSRF_COOKIE_SECURE значення True, таким чином Cookie будуть переселатися тільки по HTTPS.
- Використовувати HTTP Strict Transport Security, що дозволить зменшити ймовірність успішної атаки SSL-stripping man-in-the-middle (MITM).

3.3 Процес розробки веб-додатку використовуючи модель безпечної розробки Whitelist

В першу чергу нам потрібно змоделювати діаграму яка відображає дозволену поведінку і переходи нашого веб-застосунку. Створимо просту діаграму згідно якою будемо проектувати застосунок і визначати модель білого списку. Створена модель міститься на рисунку 3.1 нижче.

У данній схемі було зображено можливі переходи для авторизованих і для анонімних користувачів. Щоб виключити загромадження схема була спрощена, перенаправлення при виконанні операції успішно або не успішно зі схеми було виключено. Легко зрозуміти що при виконані операції на сторінці додати пост і редагування посту у разі успішного або неуспішного виконання перенаправить користувача на сторінку теми.

Рисунок 3.1 – діаграма форума яка відображає дозволену поведінку програми

Далі опишемо формальну модель нашого веб-застосунку.

Білий список {C, D, W, S} визначимо наступним чином:

Множина C – множина усіх компонентів форума і складається з:

C = (u, Reg, Login, Logout, Boards, PassChange, Account, Topics, AddTopic, Posts, EditPosts, AddPost)

Де компоненти системи мають наступне значення:

- u – зовнішні компоненти системи. Присутній для повноти моделі
- Reg – реєстрація користувачів
- Login –автенифікація користувачів
- Logout – деавтентифікація користувачів
- Account – особиста сторінка користувача
- PassChange – підсистема зміни паролю
- Boards – домашня сторінка з списком форумів
- Topics – список тем форума
- AddTopic – підсистема створення теми
- Posts – сторінка теми
- EditPost – зміна посту
- AddPost – додавання посту

Множина D – множина усіх станів компонентів форума і складається з:

D = (d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10})

Де стани системи мають наступне значення:

- d_1 – користувач анонімний
- d_2 – користувач аутентифікований
- d_3 – попереднє представлення (сторінка)
- d_4 – невдала аутентифікація
- d_5 – відміна аутентифікації
- d_6 – успішна аутентифікація
- d_7 – невдала реєстрація
- d_8 – відміна реєстрації

- d_9 – успішна реєстрація
- d_{10} – тема створена успішно

Тоді матрицю W впорядкованих пар станів можна задати наступним чином.

Таблиця 3.1 – матриця W впорядкованих пар станів переходу компонентів

	u	Reg	Login	Logout	Account	PassChange
u	-	-	-	-	-	-
Reg	-	{d1, d7}	{d1, d8}	-	-	-
Login	-	{d1, d5}	{d1, d4}	-	-	-
Logout	-	-	-	-	-	-
Account	-	-	-	{d2, d3=account}	{d2, d3=account}	{d2, d3=account}
PassChange	-	-	{d2, d4=logout}	-	-	-
Boards	-	{d1, d3=boards}	{d1, d3=boards}	{d2, d3=boards}	{d2, d3=boards}	-
Topics	-	{d1, d3=topics}	{d1, d3=topics}	{d2, d3=topics}	{d2, d3=topics}	-
AddTopic	-	-	-	{d2, d3=addtopic}	{d2, d3=addtopic}	-
Posts	-	{d1, d3=posts}	{d1, d3=posts}	{d2, d3=posts}	{d2, d3=posts}	-
EditPost	-	-	-	{d2, d3=editpost}	{d2, d3=editpost}	-
AddPost	-			{d2, d3=addpost}	{d2, d3=addpost}	-

	Boards	Topics	AddTopic	Posts	EditPost	AddPost
u	-	-	-	-	-	-
Reg	{d1, d8 or d9 }	-	-	-	-	-
Login	{d1, d5 or d6}	-	-	-	-	-
Logout	{d2, d3!=reg, login}	-	-	-	-	-
Account	{d2, d3=account}	-	-	-	-	-
PassChange	-	-	-	-	-	-
Boards	{d1 or d2, d3=boards}	{d1 or d2, d3=boards}	-	-	-	-
Topics	{d1 or d2, d3=topics}	{d1 or d2, d3=topics}	{d2, d3=topics}	{d1 or d2, d3=topics}	-	-
AddTopic	{d2, d3=addtopic}	{d2, d3=addtopic}	{d2, d3=addtopic}	{d2, d10}	-	-
Posts	{d1 or d2, d3=posts}	{d1 or d2, d3=posts}	-	{d1 or d2, d3=posts}	{d2, d3=posts}	{d2, d3=posts}
EditPost	{d2, d3=editpost}	{d2, d3=editpost}	-	{d2, d3=editpost}	{d2, d3=editpost}	-
AddPost	{d2, d3=addpost}	{d2, d3=addpost}	-	{d2, d3=addpost}	-	{d2, d3=addpost}

(Таблиця читається з колонки)

Представимо матрицю W у виді двійкової матриці, де 1 – дозволений перехід, а 0 заборонений.

Таблиця 3.2 – двійкова матриця переходу станів компонентів W

	u	Reg	Login	Logout	Account	PassChange	Boards	Topics	AddTopic	Posts	EditPost	AddPost
u	0	0	0	0	0	0	0	0	0	0	0	0
Reg	0	1	1	0	0	0	1	0	0	0	0	0
Login	0	1	1	0	0	0	1	0	0	0	0	0
Logout	0	0	0	0	0	0	1	0	0	0	0	0
Account	0	0	0	1	1	1	1	0	0	0	0	0
PassChange	0	0	1	0	0	0	0	0	0	0	0	0
Boards	0	1	1	1	1	0	1	1	0	0	0	0
Topics	0	1	1	1	1	0	1	1	1	1	0	0
AddTopic	0	0	0	1	1	0	1	1	1	1	0	0
Posts	0	1	1	1	1	0	1	1	0	1	1	1
EditPost	0	0	0	1	1	0	1	1	0	1	1	0
AddPost	0	0	0	1	1	0	1	1	0	1	0	1

Матриця у таблиці 3.2 представлена у двійковому вигляді. Що дозволить створити програмне забезпечення яке буде сканувати наш веб-застосунок і перевіряти чи перехід компонентів з одного представлення в інше дозволений. Таке програмне забезпечення буде корисним у разі створення комплексного веб-додатку, коли перевірка дозволеності переходу компонентів важко перевірятимется вручну, програма зможе контролювати легітимність пере направлення сторінок і у разі помилки повідомляти про це розробника.

Ось так виглядають впорядковані пари станів дозволених переходів у вигляді елементів множити W.

$$W = \{ (reg, reg);\ (reg, login); (reg, boards); (login, reg); (login, login);$$

$$(login, boards); (logout, boards);\ (account, logout); (account, passchange);$$

$$(account, boards);\ (passchange, login);\ (boards, reg);\ (boards, login);$$

$$(boards, logout); (boards, account); (boards, boards); (boards, topics);$$

$$(topics, reg); (topics, login); (topics, logout); (topics, account);$$

$$(topics, boards); (topics, topics); (topics, addtopic); (topics, posts);$$

$$(addtopic, logout); (addtopic, account); (addtopic, boards); (addtopic, topics);$$

$$(addtopic, addtopic); (addtopic, posts); (posts, reg); (posts, login); (posts, logout);$$

$$(posts, account); (posts, boards); (posts, topics); (posts, posts); (posts, editpost);$$

$$(posts, addpost); (editpost, logout); (editpost, account); (editpost, boards);$$

$$(editpost, topics); (editposts, posts); (editpost, editpost); (addpost, logout);$$

$$(addpost, account); (addpost, boards); (addpost, topics); (addpost, posts);$$

$$(addpost, addpost) \}$$

Далі створимо таблицю 3.3 для матриці S яка відображає безпечні переходи, куди будуть перенаправлені веб-сторінки у разі недозволеного переходу або помилки при виконані дозволеного переходу.

Таблиця 3.3 – матриця переходу веб-сторінок у безпечні стани

	u	Reg	Login	Logout	Account	PassChange
u	N/A	boards	boards	boards	boards	boards
Reg	boards	reg	boards	login	login	login
Login	boards	boards	login	login	login	login
Logout	u	boards	login	logout	login	login
Account	logout	logout	logout	logout	logout	logout
PassChange	logout	logout	logout	login	login	login
Boards	u	boards	boards	login	login	login
Topics	boards	boards	boards	login	login	login
AddTopic	boards	logout	logout	login	login	login
Posts	boards	boards	boards	login	login	login
EditPost	boards	logout	logout	login	login	login
AddPost	boards	logout	logout	login	login	login

	Boards	Topics	AddTopic	Posts	EditPost	AddPost
u	boards	boards	boards	boards	boards	boards
Reg	boards	boards	login	boards	login	login
Login	boards	boards	login	boards	login	login
Logout	boards	boards	login	boards	login	login
Account	logout	logout	logout	logout	logout	logout
PassChange	boards	boards	login	boards	login	login
Boards	boards	boards	login	boards	login	login
Topics	boards	boards	login	boards	login	login
AddTopic	boards	boards	login	boards	login	login
Posts	boards	boards	login	boards	login	login
EditPost	boards	boards	login	boards	login	login
AddPost	boards	boards	login	boards	login	login

У разі створення веб-застосунку з не громіздкою інфраструктурою таблицею матриці переходу представлень у безпечні стани користуватись зручно. У разі комплексного застосунку, існує потреба програмної реалізації для перевірки і контролю перенаправлень представлень у безпечні

Один з перших кроків створення веб-застосунку це проектування бази даних. У нашому веб-застосунку використовувалась база даних за замовчуванням – SQL Lite. З міркувань безпеки модель користувача використовувалась вбудована у django.contrib.auth. На рисунку 3.2 нижче зображена UML діаграми нашої бази даних.

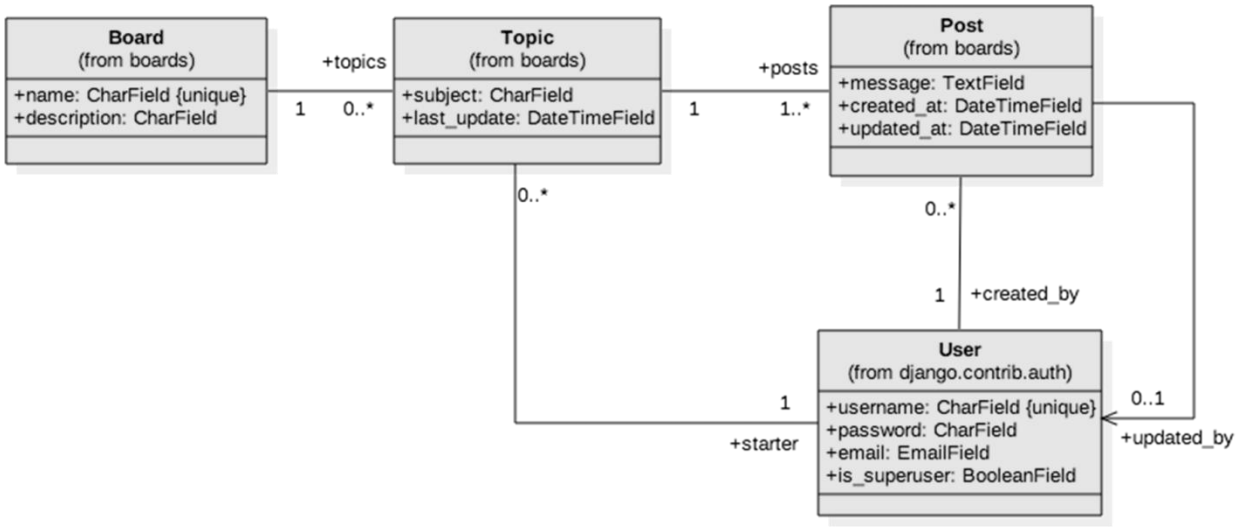

Рисунок 3.2 – автоматично згенерована UML діаграма бази данних

3.4 Використані методи забезпечення відповідності до моделі

Проаналізувавши фреймворк Django було знайдено, що найменше 2 адекватних шляхів забезпечення веб-застосунку на відповідність до створеної моделі білого списку у даній роботі, а саме перенаправлення недозволених переходів, або виникненню помилку при дозволеному переході у безпечні представлення або простими словами сторінки.

Перший шлях заключається у тому, що створити деяку програму посередник (Middleware)[12] яка буде керувати усіма переходами, визначати дозволені і заборонені а також перенаправляти на безпечні представлення. Що вприципі, добре вписується в архітектуру самого веб-каркасу Django і дозволило б

автоматизувати увесь процес і ефективно впровадити і застосовувати модель білого списку. Взаємодію складових веб-каркасу Django представлено на рисунку 3.3.

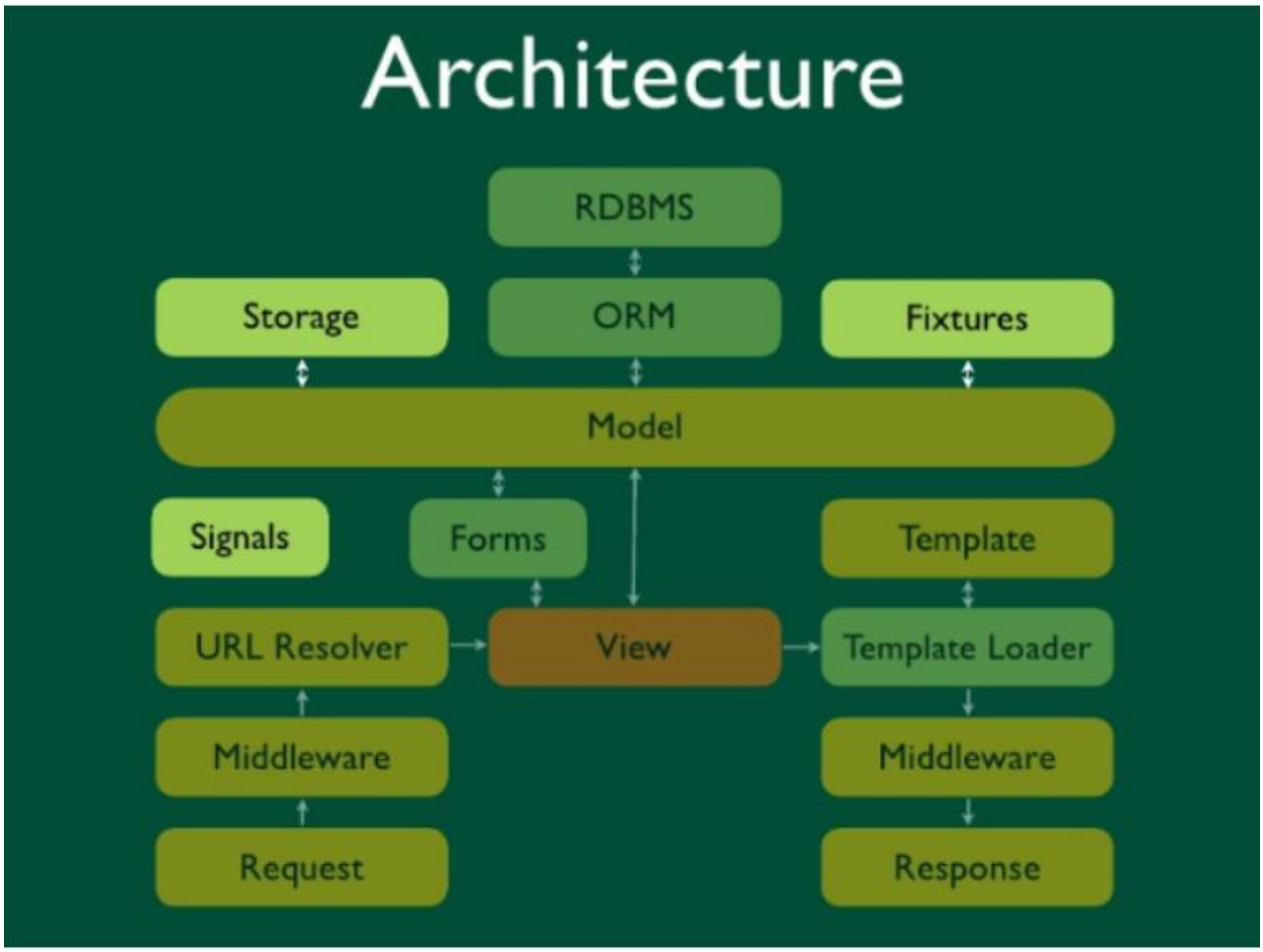

Рисунок 3.3 – Архітектура веб-каркасу Django[11]

Переваги даного методу в тому, що весь функціонал моделі білого списку можна було б забезпечити у одній бібліотеці і просто дописати цю бібліотеку у middleware. Недолік у тому, що потрібно повністю розуміти архітектуру фреймоворку на низькому рівні і продвинуті уміння кодування python. Розробка даного middleware потребує значних ресурсів.

Цей спосіб реалізації відкидається з міркувань затрат по часу і ймовірності правильної реалізації. Оскільки задача полягає в тому, щоб реалізувати і

перевірити концепцію білого списку, нам підійде більш простий але менш ефективний спосіб. На заміну йому я використаю більш простий метод який складається з деякого комплексу прийомів.

Якщо для переходу на сторінку my_account потрібна утентифікація користувача, кращий спосіб це використовувати вбудований клас користувача Django і тоді можна використати декоратор login_required, який забезпечує доступ до представлення лише якщо сесія користувача активна. Нижче наведена реалізація такого способу у представлені accounts/views.py для входу на особисту сторінку my_account.html:

```
from django.utils.decorators import method_decorator

@method_decorator(login_required, name='distpach')

class UserUpdateView():
```

Для представлення створення нової сторінки теж застосований декоратор login_required, але основна ідея заключається в тому, що якшо авторизований користувач створює нову тему і валідація форм пройшла успішно, то створюється новий об'єкт моделі topic у базі даних і далі користувач перенаправляється на сторінку створеної теми. У разі провалу валідації форми, користувач перенаправиться на ту ж саму сторінку створення теми і отримає інформацію що до правильного заповнення форми, механізм валідації форм використовувався вбудований у Django.

```
If request.method == 'POST'

If form.is_valid():

Return redirect('topic_posts')

Else:

Return render(request, 'new_topic.html')
```

Задамо безпечні переходи (login, boards), (logout, boards) шляхом запису наступного у settings.py

LOGOUT_REDIRECT_URL = ‘boards’

LOGIN_REDIRECT_URL = ‘boards’

LOGIN_URL = ‘login’

Додамо у шаблон topic_posts.html який виводить сторінку теми можливість переходу за для редагування посту тільки при умові якщо цей пост створений користувачем який на даний момент авторизований і має активний сеанс:

{% if post.created_by == user %}

Такий спосіб дозволить лише візуально відібрати можливість редагування постів які не належать користувачеві, але якщо спробувати провести атаку і в адресну строку браузеру ввести шлях до редагування посту, то через наступний фільтр queryset методу:

queryset = super().get_queryset().filter(created_by=self.request.user)

Ми вивводимо авторизованого користувача автора поста, якщо авторизований користувач не той що зазначений у автора поста. Валадація не пропустить редагування даного посту і поверне безпечне представлення списку форумів яка виведе сторінку home.html.

Для перевірки правильності усіх перенаправлень які дозволені і заборонені білим списком під час розробки було створено 67 юніт тестів у яких використовувались методи self.assertTrue self.assertRedirects self.assertContains за для перевірки допустимості переходів представлень і правильності переходу у разі спрацювання безпечного переходу.

3.5 Використаний інструмент для перевірки

Перевірка веб-застосунку на відповідність моделі білого списку буде виконуватися за допомогою інструменту для тестування веб-застосунків на вразливості OWASP ZAP.

OWASP ZAP (Zed Attack Proxy)[13] – сканер безпеки веб-застосунків який знаходиться у вільному доступі. Це один з найбільш активних проектів OWASP. OWASP ZAP має статус флагмана. Якщо використовувати ZAP як проксі-сервер він дозволяє користувачу маніпулювати трафіком який проходить через за стосунок включаючи http і https трафік. Також ZAP може працювати в режимі демону, який керується за допомогою REST API.

OWASP ZAP найбільш використовуваний вбудований функціонал:

- Проксі-сервер з можливістю перехвату трафіка: програма налаштовує VPN і проксі сервер який фіксує http/https трафік
- AJAX CRAWL (веб-паук, пошуковий бот): сканування сторінок сайту з ціллю отримати необхідну інформацію. Результат виконання даного сканування є html представлення в кінечній точці (в кожної пошукової системи свої налаштування, а саме загружати чи ні javascript, css, зображення і так далі,).
- Автоматичний сканер
- Пасивний сканер
- Інструменти примусового браузингу
- Фаззер
- Підтримка веб-сокетів
- Підтримка Plag-n-hack

ZAP побудований на архітектурі плагінів і має онлайн магазин який дозволяє використовувати новий функціонал або оновити уже існуючий. Інтерфейс користувача зручний і інтуїтивний.

В першу чергу OWASP ZAP – інструмент який досить легко використувати для тестування на проникнення (penetration testing) і пошуку вразливостей веб-застосунків. Програма створена для користувачів які вже мають досвід роботи в сфері інформаційної безпеки таких як розробники і ручні тестувальньики так і для новачків в області інформаційної безпеки, зокрема безпеки веб-застосунків.

Режими роботи OWASP ZAP:

- Безпечний режим – при використанні данного режиму неможливо зробити щось потенційно небезпечно для веб-застосунку
- Захищений режим – за допомогою цього режиму користувач (наш бот) може виконувати тільки шкідливі дії в області url адресів указаних в області браузера.
- Стандарний режим – в цьому режимі наш користувач (бот), може робити все, що дозволяє робити наш веб-застосунок
- Режим Атаки – при знаходженні нових вузлів в області дії шпигуна, вузли активну скануються, як тільки вони були знайдені

Використання сканеру:

Після того як запустите сканер, всі знайдені помилки OWASP ZAP будуть відсортовані під різні серйозності вразливостей і будуть знаходиться у вкладці «Сповіщення». Під червоний прапорець потраплять найсерйозніші, такі як XSS, SQL injaction. Під помаранчевий менш серйозні, типу CSRF і тд. Ну і інші незначні, на які мало хто звертає увагу, хоча варто було б. Приклад, рисунок 4.1.

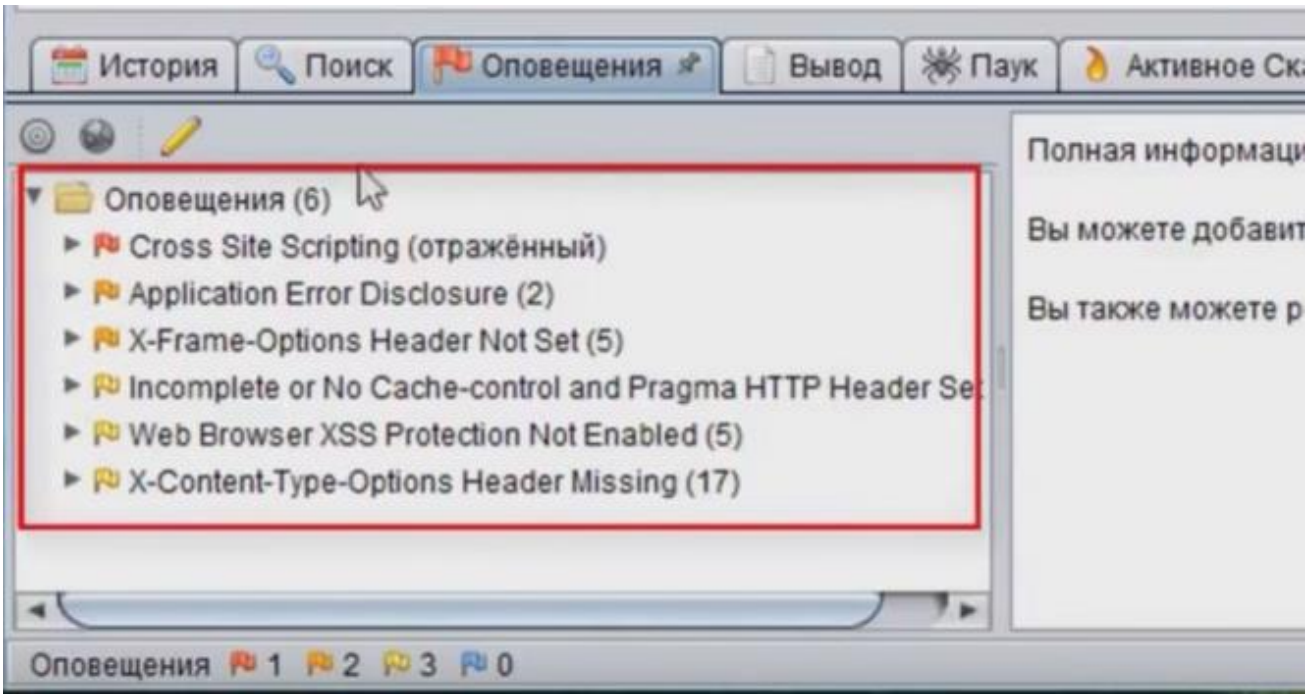

Рисунок 3.4 – відсортований звіт вразливостей по рівню небезпеки

Щоб подивитися більш детально знайдені вразливості, клікнувши кілька з якоїсь з вразливостей.

При цьому відкриється вікно в якому буде розписано наступне:

- що це за вразливість і на, що вона впливає
- її критичність
- де її можна відтворити
- можливий варіант устарения

Приклад на рисунку 3.5

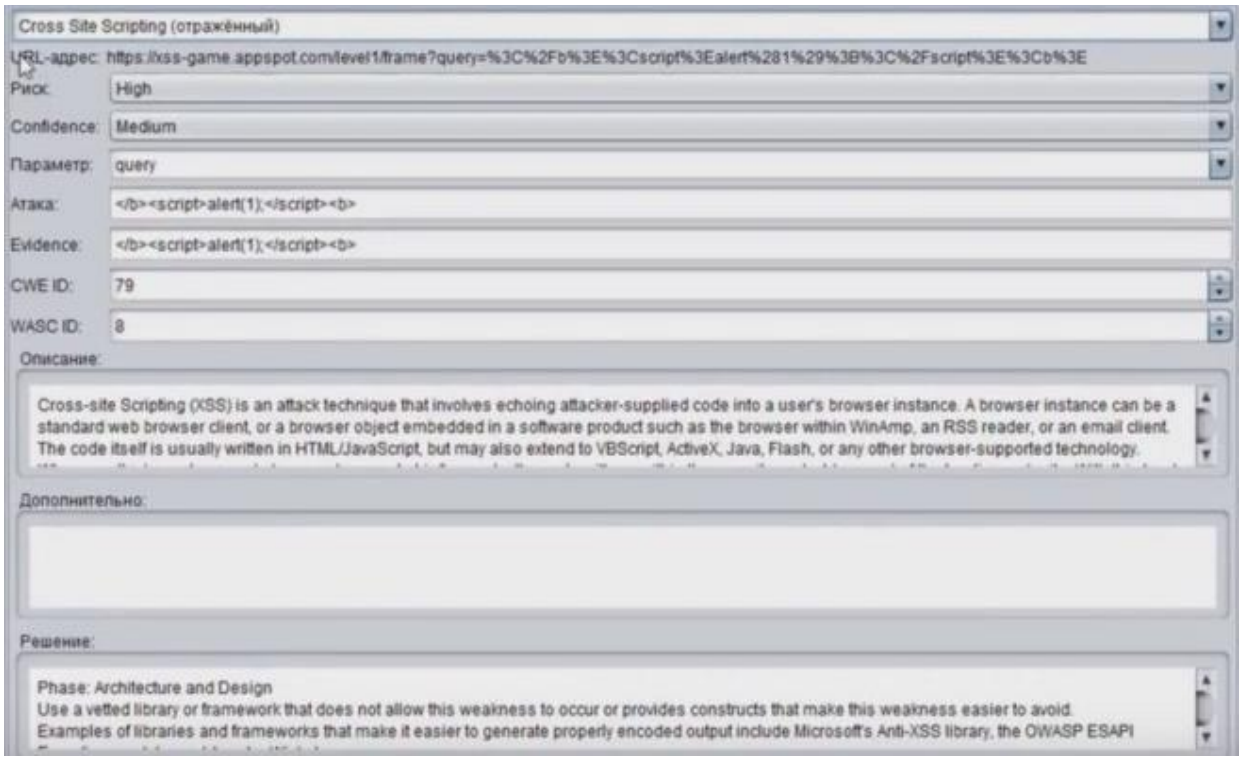

Рисунок 3.5 – детальний розгляд знайденої вразливості

3.6 Тестування за допомогою OWASP ZAP

Тестування сайту на відповідність до розробленої у даній роботі моделі білого списку виконувалось у наступній послідовності і за допомогою таких тестів як у таблиці 3.4:

Таблиця 3.4 – використані тести ZAP для дослідження

Режим атаки, стандартний режим, захищений режим		
Веб-паук	Активний режим	Пасивний режим
Активна HTTP сесія користувача	Активна HTTP сесія користувача	Активна HTTP сесія користувача
Не активна HTTP сесія користувача	Не активна HTTP сесія користувача	Не активна HTTP сесія користувача

Варіант тестування сайту з авторизованим користувачем:

За для зменшення загромадженості розділу графічне представлення кроків проведення тестування авторизованого користувача будуть винесені у додаток А.

Якщо у вашому проекті присутня авторизація, то ті посилання, які доступні для авторизованих користувачів, стануть недоступними для OWASP ZAP без налаштованого користувача, під яким зможе зайти наш застосунок для сканування гіперпосилань всередині.

Для того, щоб швидко налаштувати цього користувача, натискаємо на іконку браузера. При цьому відкриється сам браузер, в якому введені настройки проксі для прийому данних сканером OWASP ZAP.

Далі потрібно виконати авторизацію на сайті через цей браузер. Всі дані, які ви ввели будуть перехоплені сканером OWASP ZAP

Потім, щоб наш сканер не сканував все що в нього потрапить (зайві ресурси), що не стосується нашого проекту, потрібно задати "контекст" тільки для однієї папки, яку ми хочемо перевірити. Для цього вибираємо потрібну папку і включаємо її в контекст.

Після того як ми задали папці контекст, ми повинні знайти в ній запит який виконувався на авторизацію. А потім створити користувача для цього методу.

Після виконаних операцій створюємо користувача, щоб він міг авторизуватися, при скануванні нашого проекту. Для цього заходимо в розділ користувачів. І в цьому розділі додаємо нашого користувача, який зможе авторизуватися.

У відкритому вікні задаємо цьому користувачеві ім'я (не має значення яке, це робиться для вашого розуміння, що це за юзер), пароль і логін (email), який підходить для авторизації в нашу систему.

Тепер нам треба показати OWASP ZAP якийсь локатор (Xpath) на сторінці авторизованого користувача, щоб програма розуміла, що авторизація пройшла успішно. Для цього нам треба зайти на запит, який ми отримали від сервера, з html розміткою, яку отримує авторизований юзер. Потім знаходимо якийсь

локатор до якого прив'яжемо наш сканер. Цей локатор OWASP буде шукати після виконання авторизації і перевіряти успішно чи ні пройшов авторизацію на сайті.

Цей локатор вставляємо в розділ аутентифікації, в якому ми задавали параметри входу. У вхід Regex pattern indentifield in Logged

Як тільки ми виконали все, що написано вище, можемо починати атаку і перевірку на, що здатний наш веб-застосунок. Для цього натискаємо по нашій папці, де задали контекст і тиснемо сканування або атака.

Після цього в списку, що появляється у інтерфейсі користувача знаходимо нашого користувача, під яким зайде сканер.

Тепер наше тестування було запущено і ми отримали звіт вразливостей більш детальний, так як змого запустити тест у більш повному варіанті з урахуванням проблем безпеки авторизованих і не авторизованих користувачів.

Жодний метод тестування не гарантує стовідсоткової точності знаходження вразливостей.

3.7 Результати тестування

Короткі результати пасивного сканування

На рисунку 3.6 зображений короткий результат пасивного сканування. Більш детальний розгляд цих вразливостей у додатку. Проаналізувавши результат сканування зрозуміло що даний вид сканування ніяким чином не доводить чи спростовує відповідність розробленого додатку до білого списку. Лише показує деякі типові вразливості, які виникли внаслідок використання деяких параметрів безпеки за замовчуванням

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	6
Informational	0

Рисунок 3.6 – короткий звіт знайдених вразливостей пасивним скануваням

Частково довести, що веб-застосунок відповідає створеній концепції вдалося за рахунок сканування в режимі павука. Ось на рисунку 3.7 частина результату сканування з активною сесією користувача в режимі web-crawl:

Метод	URL-адрес	Код	Причина
POST	http://localhost:8000/accounts/login/	403	Forbidden
POST	http://localhost:8000/accounts/login/?next=/board/boards/2/new/	200	OK
POST	http://localhost:8000/accounts/signup/	403	Forbidden
POST	http://localhost:8000/accounts/login/?next=/board/boards/2/topics/7/reply/	200	OK
POST	http://localhost:8000/accounts/login/?next=/board/boards/3/new/	200	OK

Рисунок 3.7 – результат сканування з активною сесією користувача в режимі web-crawl

Авторизований користувач не може відправити пост запит повторної автентифікації поки його сесія активна і не перейти на сторінку реєстрації.

Інші запити у таблиці показують, що неавторизований користувач при спробі створити нову тему або добавити пост в уже існуючу перенаправляється до безпечного шляху, на сторінку авторизації

На рисунку 3.8 короткий звіт активного сканування в режимі атаки:

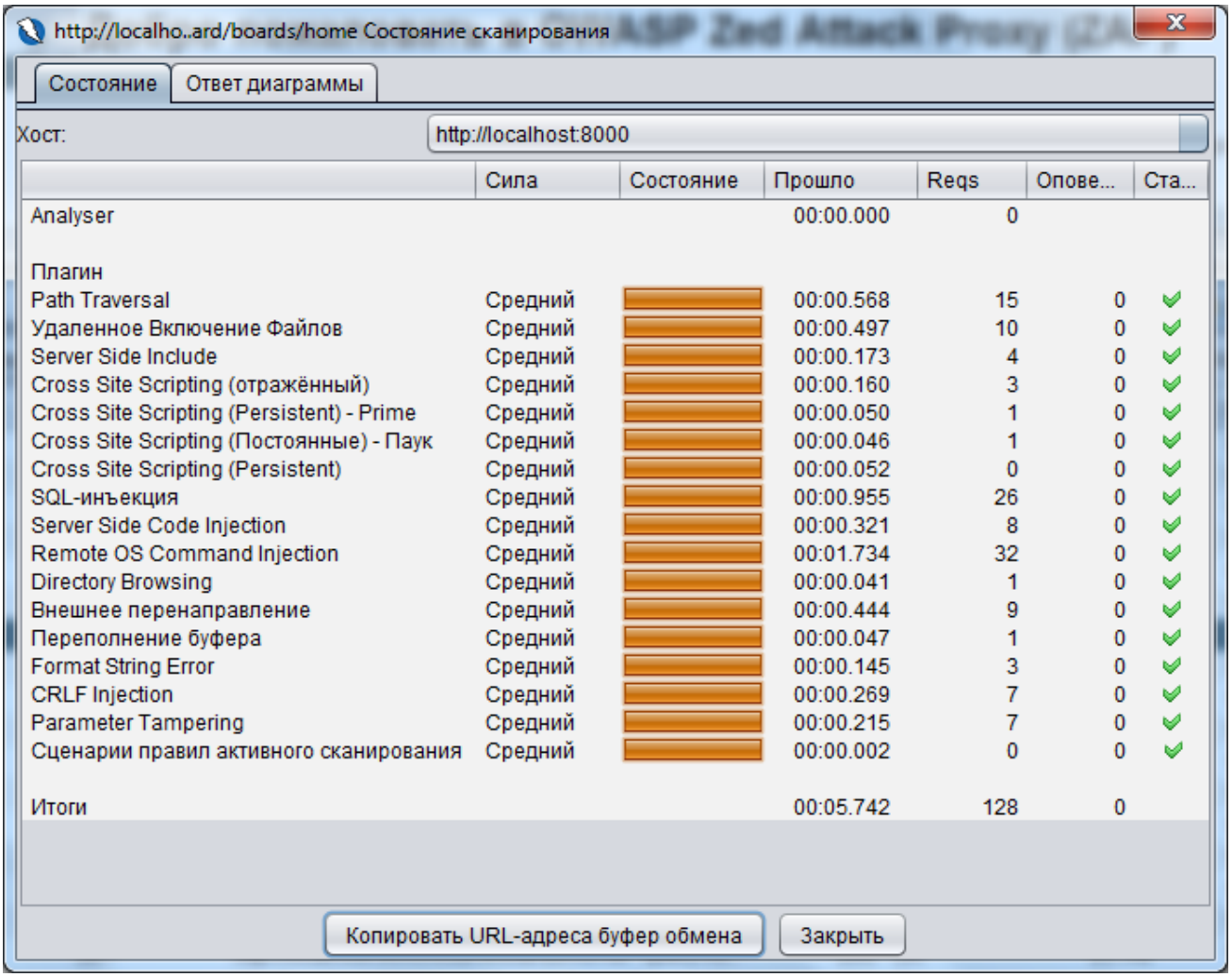

Рисунок 3.8 – короткий звіт активного сканування в режимі атаки

Сканування в цьому режимі як і в пасивному не дало ніякої інформації про відповідність веб-застосунку до моделі білого списку. Сканер знайшов деякі типові вразливості веб-сайтів, скоріше за все через те, що деякі параметру захисту Django були встановлені за замовчуванням. Але в даній роботі це не важливо.

Висновок розділу 3

У 3 розділі був створений застосунок на основі методики розробки білого списку і перевірена відповідність веб-застосунку до моделі білого списку створеного в даній роботі. Результат виявився задовільним і сканування в режимі веб-павука дало нам підтвердження відповідності веб-застосунку до моделі білого списку.Звідки можна зробити висновок, що теоретичні аспекти забезпечення

безпеки для веб-застосунків які використовують модель розробки білого списку дають результат і на практиці. Повні результати тестування знаходяться у додатку А.

ВИСНОВОКИ

У першому розділі буде проаналізовано проблема створення безпечних веб-заастосунків, деякі нюанси їх створення а також типові загрози.

В результаті дослідження ми прийшли до висновку, що рішеням даної проблеми є використання методик розробок безпечних веб-застосунків або комплексу методик.

У другому розділі було проаналізована концепія типового використання білого списку у ІТ а також розроблена методика використання даної моделі для розробки безпечних веб-застосунків.

Результатом другого розділу стала описана модель білого списку і адаптована математична модель для розробки безпечних веб-застосунків, також запропонована інструкціїя її застосування.

В результаті роботи над дипломною роботою ми одержали наступні результати:

1. Описана концепція методики розробки безпечних веб-застосунків з використанням загальної моделі білого списку.

2. Розроблений прототип веб-застосуноку з використанням методики розробки описаній в даній роботі а також розписані способи за допомогою яких реалізовувати відповідність поведінки веб-застосунку до моделі білого списку.

3. Отриманий в цілому задовільний результат застосування розробленої у дипломній роботі методики на основі аналізу тестів за стосунку.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Фреймворки в веб-разработке [Електронний ресурс] // Web Creator. – 2015. – Режим доступу до ресурсу: https://web-creator.ru/articles/about_frameworks.

2. MVC — модель-представление-контроллер [Електронний ресурс] // Web Creator. – 2014. – Режим доступу до ресурсу: https://web-creator.ru/articles/mvc.

3. Information technology — Vocabulary, 2015. – (ISO/IEC 2382:2015)

4. 97% of Java Apps Harbor a Known Security Hole [Електронний ресурс] // Fortune.com. – 2016. – Режим доступу до ресурсу:

5. State of app security 2016: Most common vulnerabilities, top trends [Електронний ресурс] // TechBeacon.com. – 2016. – Режим доступу до ресурсу: https://techbeacon.com/security/state-app-security-2016-most-common-vulnerabilities-top-trends.

6. Microsoft SDL [Електронний ресурс] – Режим доступу до ресурсу: https://www.microsoft.com/en-us/securityengineering/sdl.

7. OWASP Secure Coding Practices Quick Reference Guide. // OWASP. – 2010.

8. Report: Cenzic Application Vulnerability Trends Report: 2014 [Електронний ресурс] // Cenzic. – 2014. – Режим доступу до ресурсу: https://info.cenzic.com/2013-Application-Security-Trends-Report.html.

9. OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks, 2017. – (creative commons)

10. Django documentation [Електронний ресурс] // Django Software Foundation. – 2016. – Режим доступу до ресурсу: https://docs.djangoproject.com/en/2.2/.

11. Security in Django [Електронний ресурс] // Django Software Foundation. – 2016. – Режим доступу до ресурсу: https://docs.djangoproject.com/en/2.2/topics/security/.

12. Middleware [Електронний ресурс] // Django Software Foundation. – 2016. – Режим доступу до ресурсу: https://docs.djangoproject.com/en/2.2/topics/http/middleware/.

13. OWASP ZAP 2.6 Getting Started Guide, 2013. – (OWASP).

ДОДАТОК А

Результати пасивного сканування

Low (Medium)	Cookie No HttpOnly Flag
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	http://127.0.0.1:8000/accounts/login/?next=/board/boards/3/new/
Method	POST
Parameter	csrftoken
Evidence	Set-Cookie: csrftoken
URL	http://127.0.0.1:8000/accounts/login/?next=/board/boards/1/topics/3/reply/
Method	POST
Parameter	csrftoken
Evidence	Set-Cookie: csrftoken
URL	http://127.0.0.1:8000/accounts/login/?next=/board/boards/1/topics/3/reply/

Low (Medium)	Web Browser XSS Protection Not Enabled
Description	Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server
URL	http://127.0.0.1:8000/accounts/login/?next=/board/boards/2/new/
Method	POST
Parameter	X-XSS-Protection
URL	http://127.0.0.1:8000/board/boards/1/
Method	GET
Parameter	X-XSS-Protection
URL	http://127.0.0.1:8000/accounts/login/?next=/board/boards/1/topics/4/reply/
Method	POST
Parameter	X-XSS-Protection

Low (Medium)	Web Browser XSS Protection Not Enabled
Description	Web Browser XSS Protection is not enabled, or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server
URL	https://www.gstatic.com/chrome/intelligence/assist/ranker/models/translate/2017/03/translate_ranker_model_20170329.pb.bin
Method	GET
Parameter	X-XSS-Protection
Evidence	X-XSS-Protection: 0
Instances	1
Solution	Ensure that the web browser's XSS filter is enabled, by setting the X-XSS-Protection HTTP response header to '1'.

Low (Medium)	Password Autocomplete in Browser
Description	The AUTOCOMPLETE attribute is not disabled on an HTML FORM/INPUT element containing password type input. Passwords may be stored in browsers and retrieved.
URL	http://127.0.0.1:8000/accounts/login/?next=/board/boards/3/new/
Method	POST
Parameter	id_password
Evidence	<input type="password" name="password" class="form-control " required id="id_password">
URL	http://127.0.0.1:8000/accounts/login/?next=/board/boards/1/topics/3/reply/
Method	POST
Parameter	id_password
Evidence	<input type="password" name="password" class="form-control " required id="id_password">
URL	http://127.0.0.1:8000/accounts/login/?next=/board/boards/1/topics/3/reply/

Low (Medium)	X-Content-Type-Options Header Missing
Description	The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.
URL	http://127.0.0.1:8000/accounts/login/?next=/board/boards/1/topics/3/reply/
Method	POST
Parameter	X-Content-Type-Options
URL	http://127.0.0.1:8000/accounts/login/?next=/board/boards/1/topics/3/reply/
Method	GET
Parameter	X-Content-Type-Options
URL	http://127.0.0.1:8000/board/boards/1/topics/5/?page=1
Method	GET
Parameter	X-Content-Type-Options
URL	http://127.0.0.1:8000/accounts/login/?next=/board/boards/3/new/

Low (Medium)	Incomplete or No Cache-control and Pragma HTTP Header Set
Description	The cache-control and pragma HTTP header have not been set properly or are missing allowing the browser and proxies to cache content.
URL	https://www.gstatic.com/chrome/intelligence/assist/ranker/models/translate/2017/03/translate_ranker_model_20170329.pb.bin
Method	GET
Parameter	Cache-Control
Evidence	public, max-age=31536000
Instances	1
Solution	Whenever possible ensure the cache-control HTTP header is set with no-cache, no-store, must-revalidate; and that the pragma HTTP header is set with no-cache.
Reference	https://www.owasp.org/index.php/Session_Management_Cheat_Sheet#Web_Content_Caching
CWE Id	525
WASC Id	13
Source ID	3

Результати сканування у режимі веб-паука

Successfully,Fri Jun 07 11:04:48 EEST 2019,GET,http://localhost:8000/robots.txt,404,Not Found,15,177,2187,РќРёР·РєРёРѕ,

Successfully,Fri Jun 07 11:04:48 EEST 2019,GET,http://localhost:8000/sitemap.xml,404,Not Found,8,177,2190,РќРёР·РєРёРѕ,

Successfully,Fri Jun 07 11:04:48 EEST 2019,GET,http://localhost:8000/board/boards/home/,200,OK,39,199,4539,РќРёР·РєРёРѕ,"Form, Script"

Successfully,Fri Jun 07 11:04:48 EEST 2019,GET,http://localhost:8000/accounts/login/,200,OK,54,469,2312,РќРёР·РєРёРѕ,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:48 EEST 2019,GET,http://localhost:8000/accounts/signup/,200,OK,49,366,3171,РќРёР·РєРёРѕ,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:49 EEST 2019,GET,http://localhost:8000/board/boards/1/,200,OK,333,199,6909,РќРёР·РєРёРѕ,"Form, Script"

Successfully,Fri Jun 07 11:04:49 EEST 2019,GET,http://localhost:8000/board/boards/1/topics/5/,200,OK,613,344,14186,РќРёР·РєРёРѕ,"Form, Script, SetCookie"

Successfully,Fri Jun 07 11:04:49 EEST 2019,GET,http://localhost:8000/board/boards/2/,200,OK,530,199,2929,РќРёР·РєРёРѕ,"Form, Script"

Successfully,Fri Jun 07 11:04:49 EEST 2019,GET,http://localhost:8000/board/boards/2/topics/7/,200,OK,432,343,3939,РќРёР·РєРёРѕ,"Form, Script, SetCookie"

Successfully,Fri Jun 07 11:04:50 EEST 2019,GET,http://localhost:8000/static/css/bootstrap.min.css,200,OK,11,188,155758,РќРёР·РєРёРѕ,Comment

Successfully,Fri Jun 07 11:04:49 EEST 2019,GET,http://localhost:8000/board/boards/3/,200,OK,227,199,2153,РќРёР·РєРёРѕ,"Form, Script"

Successfully,Fri Jun 07 11:04:50 EEST 2019,GET,http://localhost:8000/static/css/navbar_font.css,200,OK,15,185,152,РќРёР·РєРёРѕ,

Successfully,Fri Jun 07 11:04:50 EEST 2019,GET,http://localhost:8000/static/css/accounts.css,200,OK,5,185,275,РќРёР·РєРёРѕ,

Successfully,Fri Jun 07 11:04:50 EEST 2019,GET,http://localhost:8000/static/js/jquery-

3.3.1.min.js,200,OK,20,189,86927,РќРёР·РєРёР№,"Form, Script, Comment"

Successfully,Fri Jun 07 11:04:50 EEST
2019,GET,http://localhost:8000/static/js/popper.min.js,200,OK,16,189,19033,РќРёР·РєРёР№,Comment

Successfully,Fri Jun 07 11:04:50 EEST
2019,GET,http://localhost:8000/static/js/bootstrap.min.js,200,OK,8,189,58072,РќРёР·РєРёР№,"Hidden, Comment"

Successfully,Fri Jun 07 11:04:50 EEST
2019,POST,http://localhost:8000/accounts/login/,403,Forbidden,9,177,2513,РќРёР·РєРёР№,

Successfully,Fri Jun 07 11:04:50 EEST
2019,POST,http://localhost:8000/accounts/signup/,200,OK,31,366,3523,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:50 EEST
2019,GET,http://localhost:8000/board/boards/1/topics/6/,200,OK,392,343,2907,РќРёР·РєРёР№,"Form, Script, SetCookie"

Successfully,Fri Jun 07 11:04:50 EEST
2019,GET,http://localhost:8000/board/boards/1/topics/6/?page=1,200,OK,47,199,2907,РќРёР·РєРёР№,"Form, Script"

Successfully,Fri Jun 07 11:04:50 EEST
2019,GET,http://localhost:8000/board/boards/1/topics/4/,200,OK,325,343,2900,РќРёР·РєРёР№,"Form, Script, SetCookie"

Successfully,Fri Jun 07 11:04:51 EEST
2019,GET,http://localhost:8000/board/boards/1/topics/4/?page=1,200,OK,36,199,2900,РќРёР·РєРёР№,"Form, Script"

Successfully,Fri Jun 07 11:04:50 EEST
2019,GET,http://localhost:8000/board/boards/1/topics/5/?page=1,200,OK,1030,200,14186,РќРёР·РєРёР№,"Form, Script"

Successfully,Fri Jun 07 11:04:51 EEST
2019,GET,http://localhost:8000/board/boards/1/topics/3/,200,OK,380,343,2932,РќРёР·РєРёР№,"Form, Script, SetCookie"

Successfully,Fri Jun 07 11:04:51 EEST
2019,GET,http://localhost:8000/board/boards/1/topics/2/,200,OK,546,343,2905,РќРёР·РєРёР№,"Form, Script, SetCookie"

Successfully,Fri Jun 07 11:04:52 EEST
2019,GET,http://localhost:8000/board/boards/1/topics/2/?page=1,200,OK,37,199,2905,РќРёР·РєРёР№,"Form, Script"

Successfully,Fri Jun 07 11:04:51 EEST
2019,GET,http://localhost:8000/board/boards/1/topics/3/?page=1,200,OK,798,199,2932,РќРёР·РєРёР№,"Form, Script"

Successfully,Fri Jun 07 11:04:52 EEST

2019,GET,http://localhost:8000/board/boards/1/topics/1/,200,OK,322,343,2907,РќРёР·РєРёР№,"Form, Script, SetCookie"

Successfully,Fri Jun 07 11:04:52 EEST 2019,GET,http://localhost:8000/board/boards/1/topics/1/?page=1,200,OK,780,343,2907,РќРёР·РєРёР№,"Form, Script, SetCookie"

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/board/boards/2/topics/7/?page=1,200,OK,34,199,3939,РќРёР·РєРёР№,"Form, Script"

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/accounts/login/?next=/board/boards/1/new/,200,OK,39,469,2332,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,POST,http://localhost:8000/accounts/signup/,200,OK,32,366,3523,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/accounts/login/?next=/board/boards/2/new/,200,OK,33,469,2332,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/5/reply/,200,OK,21,469,2343,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/accounts/login/?next=/board/boards/2/topics/7/reply/,200,OK,29,469,2343,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/accounts/login/?next=/board/boards/3/new/,200,OK,31,469,2332,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/6/reply/,200,OK,28,469,2343,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,POST,http://localhost:8000/accounts/signup/,200,OK,36,366,3523,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/3/reply/,200,OK,28,469,2343,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/4/reply/,200,OK,16,469,2343,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/2/reply/,200,OK,20,469,2343,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/1/new/,200,OK,357,469,2537,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/1/reply/,200,OK,21,469,2343,РќРё Р·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/2/new/,200,OK,394,469,2537,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:53 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/5/reply/,200,OK,385,469,2548,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:54 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/2/topics/7/reply/,200,OK,420,469,2548,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:54 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/3/new/,200,OK,358,469,2537,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Max Depth,Fri Jun 07 11:04:54 EEST 2019,POST,http://localhost:8000/accounts/signup/,200,OK,21,366,3523,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Max Depth,Fri Jun 07 11:04:54 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/6/reply/,200,OK,288,469,2548,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Max Depth,Fri Jun 07 11:04:54 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/3/reply/,200,OK,270,469,2548,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Max Depth,Fri Jun 07 11:04:54 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/4/reply/,200,OK,268,469,2548,РќРёР·РєРё

Р№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Max Depth,Fri Jun 07 11:04:54 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/2/reply/,200,OK,266,469,2548,РќРёР·РєРё Р№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Max Depth,Fri Jun 07 11:04:54 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/1/new/,200,OK,266,469,2537,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Max Depth,Fri Jun 07 11:04:55 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/1/reply/,200,OK,304,469,2548,РќРёР·РєРё Р№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Max Depth,Fri Jun 07 11:04:55 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/2/new/,200,OK,300,469,2537,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Max Depth,Fri Jun 07 11:04:55 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/1/topics/5/reply/,200,OK,350,469,2548,РќРёР·РєРё Р№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Max Depth,Fri Jun 07 11:04:55 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/2/topics/7/reply/,200,OK,300,469,2548,РќРёР·РєРё Р№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Max Depth,Fri Jun 07 11:04:55 EEST 2019,POST,http://localhost:8000/accounts/login/?next=/board/boards/3/new/,200,OK,258,469,2537,РќРёР·РєРёР№,"Form, Password, Hidden, Script, SetCookie, AntiCSRF"

Successfully,Fri Jun 07 11:04:50 EEST 2019,GET,http://localhost:8000/board/boards/1/new/,302,Found,24,253,0,,

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/board/boards/2/new/,302,Found,6,253,0,,

Successfully,Fri Jun 07 11:04:52 EEST 2019,GET,http://localhost:8000/board/boards/1/topics/5/reply/,302,Found,488,264,0,,

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/board/boards/2/topics/7/reply/,302,Found,12,264,0,,

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/board/boards/3/new/,302,Found,15,253,0,,

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/board/boards/1/topics/6/reply/,302,Found,7,264,0,,

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/board/boards/1/topics/4/reply/,302,Found,12,264,0,,

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/board/boards/1/topics/3/reply/,302,Found,16,264,0,,

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/board/boards/1/topics/2/reply/,302,Found,7,264,0,,

Successfully,Fri Jun 07 11:04:53 EEST 2019,GET,http://localhost:8000/board/boards/1/topics/1/reply/,302,Found,20,264,0,,

Результати активного сканування

ID,Req. Timestamp,Resp. Timestamp,РњРµС‚РѕРґ,URL-Р°РґСЂРµСЃ,РљРѕРґ СЃС‚Р°С‚СѓСЃР°,RTT,Size Resp. Header,Size Resp. Body

693,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=c%3A%2FWindows%2Fsystem.ini,200,OK,38,199,4539

694,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2FWindows%2Fsystem.ini,200,OK,34,199,4539

695,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=c%3A%5CWindows%5Csystem.ini,200,OK,34,199,4539

696,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=..%5C..%5C..%5C..%5C..%5C..%5C..%5C..%5C..%5C..%5C..%5C..%5C..%5C..%5C..%5C..%5C..%5CWindows%5Csystem.ini,200,OK,22,199,4539

697,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%2Fetc%2Fpasswd,200,OK,27,199,4539

698,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd,200,OK,23,199,4539

699,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=c%3A%2F,200,OK,24,199,4539

700,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%2F,200,OK,25,199,4539

701,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST

2019,GET,http://localhost:8000/board/boards/home/?query=c%3A%5C,200,OK,34,199,4539

702,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F,200,OK,25,199,4539

703,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=WEB-INF%2Fweb.xml,200,OK,26,199,4539

704,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=WEB-INF%5Cweb.xml,200,OK,22,199,4539

705,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%2FWEB-INF%2Fweb.xml,200,OK,18,199,4539

706,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%5CWEB-INF%5Cweb.xml,200,OK,25,199,4539

707,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=thishouldnotexistandhopefullyitwillnot,200,OK,40,199,4539

708,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=http%3A%2F%2Fwww.google.com%2F,200,OK,29,199,4539

709,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=http%3A%2F%2Fwww.google.com%3A80%2F,200,OK,23,199,4539

710,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=http%3A%2F%2Fwww.google.com,200,OK,26,199,4539

711,Fri Jun 07 11:04:58 EEST 2019,Fri Jun 07 11:04:58 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=http%3A%2F%2Fwww.google.com%2Fsearch%3Fq%3DOWASP%2520ZAP,200,OK,20,199,4539

712,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=http%3A%2F%2Fwww.google.com%3A80%2Fsearch%3Fq%3DOWASP%2520ZAP,200,OK,20,199,4539

713,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=www.google.com%2F,200,OK,33,199,4539

714,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=www.google.com%3A80%2F,200,OK,20,199,4539

715,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST

2019,GET,http://localhost:8000/board/boards/home/?query=www.google.com,200,OK,22,199,4539

716,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=www.google.com%2Fsearch%3Fq%3DOWASP%2520ZAP,200,OK,22,199,4539

717,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=www.google.com%3A80%2Fsearch%3Fq%3DOWASP%2520ZAP,200,OK,18,199,4539

718,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E,200,OK,30,199,4539

719,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%22%3E%3C%21--%23EXEC+cmd%3D%22ls+%2F%22--%3E%3C,200,OK,38,199,4539

720,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E,200,OK,32,199,4539

721,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%22%3E%3C%21--%23EXEC+cmd%3D%22dir+%5C%22--%3E%3C,200,OK,30,199,4539

722,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=0W45pz4p,200,OK,24,199,4539

723,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query0W45pz4p,200,OK,27,199,4539

724,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%27%22%3Cscript%3Ealert%281%29%3B%3C%2Fscript%3E,200,OK,27,199,4539

725,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=zApPX2sS,200,OK,46,199,4539

726,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/,200,OK,42,199,4539

727,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%27,200,OK,26,199,4539

728,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST

2019,GET,http://localhost:8000/board/boards/home/?query=query%27,200,OK,27,199,4539

729,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:04:59 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%22,200,OK,31,199,4539

730,Fri Jun 07 11:04:59 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22,200,OK,23,199,4539

731,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%3B,200,OK,31,199,4539

732,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%3B,200,OK,31,199,4539

733,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%29,200,OK,28,199,4539

734,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%29,200,OK,34,199,4539

735,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/,200,OK,26,199,4539

736,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/,200,OK,27,199,4539

737,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query+AND+1%3D1+--+,200,OK,25,199,4539

738,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query+AND+1%3D2+--+,200,OK,29,199,4539

739,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query+OR+1%3D1+--+,200,OK,18,199,4539

740,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query+AND+1%3D2+--+,200,OK,17,199,4539

741,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query+OR+1%3D1+--+,200,OK,19,199,4539

742,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27+AND+%271%27%3D%271%27+--+,200,OK,17,199,4539

743,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27+AND+%271%27%3D%272%27+--

+,200,OK,19,199,4539

744,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27+OR+%271%27%3D%271%27+--+,200,OK,18,199,4539

745,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27+AND+%271%27%3D%272%27+--+,200,OK,19,199,4539

746,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27+OR+%271%27%3D%271%27+--+,200,OK,18,199,4539

747,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query+UNION+ALL+select+NULL+--+,200,OK,19,199,4539

748,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27+UNION+ALL+select+NULL+--+,200,OK,22,199,4539

749,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22+UNION+ALL+select+NULL+--+,200,OK,19,199,4539

750,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%29+UNION+ALL+select+NULL+--+,200,OK,18,199,4539

751,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27%29+UNION+ALL+select+NULL+--+,200,OK,17,199,4539

752,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/,200,OK,18,199,4539

753,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%22%3Bprint%28chr%28122%29.chr%2897%29.chr%28112%29.chr%2895%29.chr%28116%29.chr%28111%29.chr%28107%29.chr%28101%29.chr%28110%29%29%3B%24var%3D%22,200,OK,18,199,4539

754,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%27%3Bprint%28chr%28122%29.chr%2897%29.chr%28112%29.chr%2895%29.chr%28116%29.chr%28111%29.chr%28107%29.chr%28101%29.chr%28110%29%29%3B%24var

%3D%27,200,OK,18,199,4539

755,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%24%7B%40print%28chr%28122%29.chr%2897%29.chr%28112%29.chr%2895%29.chr%28116%29.chr%28111%29.chr%28107%29.chr%28101%29.chr%28110%29%29%7D,200,OK,19,199,4539

756,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%24%7B%40print%28chr%28122%29.chr%2897%29.chr%28112%29.chr%2895%29.chr%28116%29.chr%28111%29.chr%28107%29.chr%28101%29.chr%28110%29%29%7D%5C,200,OK,20,199,4539

757,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%3Bprint%28chr%28122%29.chr%2897%29.chr%28112%29.chr%2895%29.chr%28116%29.chr%28111%29.chr%28107%29.chr%28101%29.chr%28110%29%29%3B,200,OK,20,199,4539

758,Fri Jun 07 11:05:00 EEST 2019,Fri Jun 07 11:05:00 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%22%2Bresponse.write%28%5B100%C2%A0000*100%C2%A0000%29%2B%22,200,OK,25,199,4539

759,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=%2Bresponse.write%28%7B0%7D*%7B1%7D%29%2B,200,OK,23,199,4539

760,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=response.write%28100%C2%A0000*100%C2%A0000%29,200,OK,25,199,4539

761,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%26cat+%2Fetc%2Fpasswd%26,200,OK,22,199,4539

762,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%3Bcat+%2Fetc%2Fpasswd%3B,200,OK,20,199,4539

763,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22%26cat+%2Fetc%2Fpasswd%26%22,200,OK,20,199,4539

764,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22%3Bcat+%2Fetc%2Fpass wd%3B%22,200,OK,20,199,4539

765,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27%26cat+%2Fetc%2Fpasswd%26%27,200,OK,19,1

99,4539

766,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27%3Bcat+%2Fetc%2Fpasswd%3B%27,200,OK,19,199,4539

767,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%26sleep+15%26,200,OK,19,199,4539

768,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%3Bsleep+15%3B,200,OK,20,199,4539

769,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22%26sleep+15%26%22,200,OK,18,199,4539

770,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22%3Bsleep+15%3B%22,200,OK,21,199,4539

771,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27%26sleep+15%26%27,200,OK,18,199,4539

772,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27%3Bsleep+15%3B%27,200,OK,52,199,4539

773,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%26type+%25SYSTEMROOT%25%5Cwin.ini,200,OK,29,199,4539

774,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%7Ctype+%25SYSTEMROOT%25%5Cwin.ini,200,OK,18,199,4539

775,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22%26type+%25SYSTEMROOT%25%5Cwin.ini%26%22,200,OK,18,199,4539

776,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22%7Ctype+%25SYSTEMROOT%25%5Cwin.ini,200,OK,17,199,4539

777,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27%26type+%25SYSTEMROOT%25%5Cwin.ini%26%27,200,OK,18,199,4539

778,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:01 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27%7Ctype+%25SYSTEMROOT%25%5Cwin.ini,20

0,OK,53,199,4539

779,Fri Jun 07 11:05:01 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%26timeout+%2FT+15,200,OK,29,199,4539

780,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%7Ctimeout+%2FT+15,200,OK,23,199,4539

781,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22%26timeout+%2FT+15%26%22,200,OK,22,199,4539

782,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22%7Ctimeout+%2FT+15,200,OK,20,199,4539

783,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27%26timeout+%2FT+15%26%27,200,OK,18,199,4539

784,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27%7Ctimeout+%2FT+15,200,OK,19,199,4539

785,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%3Bget-help,200,OK,37,199,4539

786,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22%3Bget-help,200,OK,18,199,4539

787,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27%3Bget-help,200,OK,19,199,4539

788,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%3Bget-help+%23,200,OK,34,199,4539

789,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%3Bstart-sleep+-s+15,200,OK,19,199,4539

790,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%22%3Bstart-sleep+-s+15,200,OK,20,199,4539

791,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%27%3Bstart-sleep+-s+15,200,OK,25,199,4539

792,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=query%3Bstart-sleep+-s+15+%23,200,OK,28,199,4539

793,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST

2019,GET,http://localhost:8000/board/boards/home/,200,OK,23,199,4539

794,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=848628811145431108.owasp.org,200,OK,25,199,4539

795,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=http%3A%2F%2F848628811145431108.owasp.org,200,OK,25,199,4539

796,Fri Jun 07 11:05:02 EEST 2019,Fri Jun 07 11:05:02 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=https%3A%2F%2F848628811145431108.owasp.org,200,OK,24,199,4539

797,Fri Jun 07 11:05:03 EEST 2019,Fri Jun 07 11:05:03 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=http%3A%5C%5C848628811145431108.owasp.org,200,OK,23,199,4539

798,Fri Jun 07 11:05:03 EEST 2019,Fri Jun 07 11:05:03 EEST 2019,GET,http://localhost:8000/board/boards/home/?query=https%3A%5C%5C848628811145431108.owasp.org,200,OK,22,199,4539