УДК 004.9

Г. В. Олійник[1], В. А. Литвинов[2], С. В. Грибков[1]
[1] Національний університет харчових технологій,
[2] Інститут проблем математичних машин і систем Національної академії наук України

ОБРАННЯ ПРОГРАМНОЇ ПЛАТФОРМИ ДЛЯ ПОБУДОВИ МОДУЛЯ БЕЗПЕКИ WEB-ОРІЄНТОВАНОЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Розглянуто проблему забезпечення захисту web-орієнтованої системи підтримки прийняття рішень під час планування виконання договорів для підприємств, що діють у сфері інформаційних технологій. Досліджено та проведено порівняльний аналіз програмних платформ для реалізації модуля захисту розроблюваної системи. За результатами дослідження була обрана програмна платформа Spring Security, яка надає широкі можливості для створення модуля безпеки системи підтримки прийняття рішень, дає змогу реалізувати багаторівневий механізм захисту при аутентифікації та авторизації користувачів, забезпечує захист від поширених типів мережевих атак.

Ключові слова: програмна платформа, модуль безпеки, захист інформації, web-орієнтована система.

The problem of protection of web-based system of decision support in planning the implementation of contracts for companies operating in the field of information technology was considered. Comparative analysis of frameworks for the implementation of the developed system security module was conducted and investigated. According to the research findings Spring Security framework was elected. It provides opportunities for creating security module of decision support system, implementing multi-protection mechanism for users' authentication and authorization, provides protection against common types of network attacks.

Key words: framework, security module, information protection, web-based information system.

Вступ

Економічна конкуренція на сучасному ринку вимагає швидкого реагування на зміни у потребах і вимогах клієнтів, а втрата хоча б одного з них може призвести до негативних наслідків для рейтингу підприємства. Для ефективного забезпечення функціонування бізнес-процесів підприємств, діяльність яких пов'язана з наданням різноманітних інформаційних послуг за укладеними договорами, дуже важливо забезпечити інформаційну й інтелектуальну підтримку процесу прийняття рішень під час планування виконання робіт. Складність при цьому полягає у тому, що будь-які зміни можуть значно впливати на поточні графіки роботи підрозділів підприємства або інших виконавців, оскільки сформований план виконання договорів регламентує усі індивідуальні завдання в усіх ланках управління та виконання. Формування наближеного до оптимального розкладу виконання робіт для кожного замовника є нетривіальним та складним завданням, а його розв'язок потребує застосування сучасних методів оптимізації та використання інформаційно-програмних комплексів [1]. Важливо також зазначити, що замовник повинен мати постійний доступ для моніторингу виконання замовлення, а у разі потреби – вносити певні корективи, які необхідно узгоджувати з наявним станом виконання робіт. Враховуючи вищезазначене, актуальним є завдання розробки web-орієнтованої системи підтримки прийняття рішень під час планування виконання договорів.

Створювана web-орієнтована система повинна забезпечувати можливість виконання основних завдань прийняття рішень, а також їх підтримку необхідною інформацією у будь-якому місці, де є доступ до мережі Інтернет. Проте особливо важливо відзначити наявність багатьох проблем, пов'язаних з організацією захисту даних, централізованим управлінням інформаційними ресурсами, розмежуванням доступу до таких ресурсів, управлінням сеансами доступу тощо.

Саме тому виникло завдання обрання програмної платформи для побудови модуля безпеки web-орієнтованої системи підтримки прийняття рішень під час планування виконання договорів.

Аналіз досліджень та публікацій

Одним з найважливіших завдань у сучасному світі є захист інформації у корпоративних інформаційних системах, що зумовлено стрімким розвитком способів і підходів несанкціонованого доступу та модифікації. У [2] розглядаються питання інформаційної безпеки та захисту даних, зокрема в інформаційно-обчислювальних системах та мережах. У [3] розглядається проблематика оцінки рівня захисту комп'ютерних систем та мереж з урахуванням їх архітектури. В [4] описано особливості захисту інформаційних ресурсів у корпоративних мережах та системах, розглянуто основні стандарти захисту інформаційних систем, а також запропоновано підхід щодо їх оцінки. У [5] розглянуто основні аспекти забезпечення доступу до інформаційних ресурсів комп'ютерних систем, а саме: детально розглянуті способи забезпечення та реалізації аутентифікації. У [6] розглядається проблема захисту інформаційного забезпечення інформаційної системи поліграфічного підприємства, а також порівнюються програмні платформи Spring Security та Apache Shiro з метою використання їх для побудови модуля безпеки системи.

Ці публікації недостатньо досліджують деталі реалізації окремих функціональних потреб інформаційних систем. Крім того, детально не розглядається практичне застосування засобів організації безпеки для складних корпоративних систем, для яких необхідна взаємодія із сторонніми системами.

Формулювання цілей роботи

Необхідно виявити ключові особливості захисту web-орієнтованої системи підтримки прийняття рішень під час планування виконання договорів для підприємств, що діють у сфері інформаційних технологій. Провести дослідження та аналіз програмних платформ для створення модуля безпеки системи підтримки прийняття рішень. Обрати програмну платформу, яка б реалізувала багаторівневий механізм захисту під час аутентифікації та авторизації користувачів, а також забезпечувала б захист від поширених типів мережевих атак.

Особливості забезпечення захисту створюваної web-орієнтованої системи підтримки прийняття рішень

Web-орієнтована система підтримки прийняття рішень призначена для підприємств, діяльність яких пов'язана з наданням широкого спектра різнопланових послуг за укладеними договорами. Кожен договір розподілений на окремі етапи, що можуть залежно від виду робіт, паралельно або послідовно виконуватись різними підрозділами підприємства чи фірмами-партнерами. Однією з головних функцій системи є формування плану виконання для майбутніх договорів так, щоб здійснення їх етапів не порушувало наявної узгодженості проведення робіт, а завантаження за підрозділами було приблизно рівномірним. Крім того, призначення системи полягає у забезпеченні інформаційної підтримки бізнес-процесів управління підприємством, а також у наданні клієнтам інформації про процес виконання робіт за укладеними договорами. Потреба у такій функціональності системи виникла тому, що клієнти вже під час попередніх обговорень вимагають знати терміни, у які замовлення буде виконане, а під час реалізації – графік та поточний стан виконання кожного виду робіт. Користувачами цієї системи є працівники, виконавці-субпідрядники та клієнти підприємства, доступ до функцій системи для яких здійснюється через мережу Internet. Усі користувачі системи, як правило, знаходяться на значній відстані один від одного, оскільки окремі підрозділи підприємства, фірми-підрядники, а особливо офіси клієнтів, розташовані у різних кварталах, містах, навіть країнах. Кожен користувач залежно

від ролі може використовувати певний набір функцій та має обмежений доступ до визначеного переліку даних.

Важливою вимогою до безпеки системи є відокремленість засобів її реалізації від бізнес-логіки та мінімальний вплив на основний функціонал.

Враховуючи вищенаведене, реалізацію захисту системи та забезпечення розподіленого доступу доцільно здійснити за рахунок створення окремого, незалежного від інших компонентів, модуля безпеки.

На модуль безпеки створюваної web-орієнтованої системи підтримки прийняття рішень покладаються такі функції: ідентифікація та аутентифікація користувачів системи; авторизація користувачів і забезпечення доступу лише до визначених функцій, відповідно до їх ролі; управління сесіями; аудит діяльності користувачів системи з подальшим її аналізом [5]. Також необхідно відзначити, що модуль безпеки повинен: надавати можливість побудови та функціонування складної логіки розподілення доступу до функцій системи та інформації, яку вона надає; здійснювати шифрування даних користувачів під час їх передачі; забезпечувати баланс між навантаженням та стабільністю роботи системи; не порушувати бізнес-логіки виконання функцій системи; мати можливість перешкоджати мережевим атакам (наприклад, Cross-Site scripting – міжсайтовий скриптинг, Cross-Site request forgery – міжсайтова підробка запиту); здійснювати захист системи відповідно до провідних галузевих стандартів та сертифікатів.

Огляд програмних платформ забезпечення безпеки

Для створення серверної частини системи була обрана мова програмування Java. Функціональність системи побудована з використанням програмної платформи Spring, а логіку взаємодії з клієнтською частиною здійснено на основі Spring MVC. Застосування шаблону проектування MVC зумовлене вимогою відокремлення бізнес-логіки від відображення даних для кінцевого користувача та архітектурного вирішення щодо консолідації функцій забезпечення бізнес-процесів на серверній частині. Тому автори розглядали такі програмні платформи для побудови модуля безпеки: JAAS, Apache Shiro, Spring Security.

JAAS (Java Authentication and Authorization Service) є стандартною програмною платформою Java SE, що інтегрована безпосередньо у Java Development Kit і призначена для реалізації функцій аутентифікації та авторизації користувачів. Підтримується аутентифікація на основі стандартних протоколів та сертифікатів. Оскільки JAAS є вбудованою платформою з низькорівневим API (Application Programming Interface) інтерфейсом прикладного програмування, а стандартні засоби інтеграції з сервлетами та Java EE (Enterprise Edition) – платформою для розробки корпоративного програмного забезпечення з додатками є відсутні, то це обмежує можливість її ефективного використання для створення складних промислових систем. Необхідно зазначити, що побудова модуля безпеки виключно на основі JAAS може тривати доволі довго та вимагатиме підключення додаткових програмних платформ і бібліотек. Цю платформу доцільно використовувати у системах, де відсутня складна бізнес-логіка. Крім того, JAAS доцільно використовувати як базовий захист на примітивному рівні, а побудову модуля захисту покласти в іншу програмну платформу.

Apache Shiro є програмною платформою, що дає змогу розробнику забезпечити розподілений доступ до інформації, компонентів системи та інформаційних джерел, який підтримує різні варіанти зберігання аутентифікаційних даних, а також роботу через різні мережеві протоколи прикладного рівня [7]. Ця платформа забезпечує реалізацію функцій аутентифікації, ідентифікації, керування подіями та контролем доступу, управління сесіями, використання криптографічних функцій тощо. До основних недоліків платформи належать певні складнощі: за інтеграції з системою, створеною з використанням програмної платформи Spring; під час використання протоколу авторизації OAuth, що уможливлює відкривати обмежений доступ до захищених ресурсів без необхідності передачі аутентифікаційних даних користувача, якому вони належать.

Spring Security – програмна платформа, до складу якої входять гнучкі та потужні механізми для організації безпеки будь-якого рівня складності [8, 9]. Ця платформа дає можливість

реалізувати аутентифікацію та авторизацію, підтримує низку протоколів і способів їх налаштування, дає змогу створити конфігурацію з використанням кількох підходів – описувати в xml-файлі або створювати Java-класи з власною логікою обробки. Spring Security дає можливість реалізувати потужну модель безпеки без будь-якого впливу на бізнес-логіку системи та легко адаптується до специфічних вимог [10]. Це досягається за рахунок використання підходів аспектно-орієнтованого програмування, що дають змогу здійснювати додаткові операції під час виконання конкретних методів чи модулів, при цьому не змінюючи їх функціонала. Отже, є можливість управляти усіма етапами виконання транзакцій та операціями, пов'язаними з безпекою. Окрім того, надаються механізми контролю усіх, навіть дрібних операцій, що виконуються, а у разі необхідності існує можливість здійснювати додаткову перевірку легітимності дій з метою захисту даних. Доступ для користувачів надається не тільки на основі ролей, а також з використанням списку контролю доступу (Access Control List). Spring Security надає можливість використовувати такі технології для забезпечення захисту: x.509 Certificates, LDAP, Kerberos, CA Sitemender, HTTP Basic, HTTP Digest та ін. Ця програмна платформа підтримує можливості інверсії управління (Inversion of Control), що включає основні абстрактні принципи та набір рекомендацій для формування слабко зв'язаного програмного коду. Принципи інверсії управління полягають у тому, що кожен компонент системи повинен бути максимально ізольований від інших, а під час роботи не залежати від конкретної реалізації інших компонентів. В такий спосіб буде забезпечена гнучкість модуля захисту системи.

Розробляючи корпоративну систему, важливо звернути увагу на чотири основні проблеми безпеки: аутентифікацію; безпеку web-запитів; безпеку сервісного рівня системи, а саме – методів, що забезпечують безпосередню реалізацію бізнес-логіки; безпеку об'єктів доменного рівня (різні доменні об'єкти з різними рівнями доступу).

Специфікація сервлетів надає підхід до перевірки аутентифікаційних даних. Проте для цього необхідно здійснювати специфічне налаштування контейнера сервлетів та редагування його параметрів, що унеможливлює універсальність конфігурації, особливо у разі необхідності створення власного Java-класу для реалізації інтерфейсу аутентифікації. Spring Security дає змогу досягати повної незалежності конфігурації, навіть безпосередньо на рівні WAR (Web Application Archive – формат файла, що описує, як саме web-додаток запаковується до специфікації Java-сервлетів у файл формату JAR або ZIP). Крім того, Spring Security має кілька дієвих механізмів аутентифікації, що дають змогу обирати підхід до аутентифікації у різних зовнішніх середовищах під час розгортання додатка.

У специфікації сервлетів передбачена реалізація функцій захисту URI (Uniform Resource Identifier – уніфікований ідентифікатор ресурсу) запиту. За специфікацією URI може бути представлений лише в обмеженому форматі, а тому будуть використані можливості Spring Security, які забезпечать: задавання шляхів з використанням шаблону Ant або регулярних виразів; розгляд окремих частин URI (наприклад, HTTP GET-параметри); реалізацію власного джерела конфігурування даних, що дає змогу динамічно змінювати web-запит під час фактичного виконання web-додатка.

Відсутність підтримки механізмів захисту безпеки сервісного рівня системи та доменних об'єктів у стандартній специфікації сервлетів призводить до доволі серйозних обмежень під час розроблення комплексних систем. Як правило, розробники або ігнорують ці вимоги, або реалізують логіку безпеки у коді MVC-контролера і навіть всередині представлень. Такий підхід має кілька значних недоліків: авторизація є загальною потребою для безпеки додатка, а об'єднання логіки авторизації з MVC-контролерами або представленнями призводять до ускладнень під час модульного тестування та відлагодження, а також до необхідності дублювання коду, що, своєю чергою, порушує SOLID (S – single responsibility, принцип єдиного обов'язку; O – open-closed, принцип відкритості/закритості; L – Liskov substitution, принцип підстановки Барбари Лісков; I – interface segregation, принцип розділення інтерфейсу; D – dependency inversion, принцип інверсії залежностей) принципи об'єктно-орієнтованого програмування, та негативно впливає на модульність і загальну архітектуру системи; у разі необхідності створення нового типу клієнтського

додатка виникає проблема неможливості повторного використання коду авторизації, вбудованого на web-рівні. Отже, логіку авторизації доцільно вводити саме на сервісному рівні для підтримання різних типів клієнтських додатків.

У разі створення простих додатків специфікації безпеки сервлетів може бути достатньо. Проте, якщо взяти до уваги необхідність в універсальності конфігурації, обмеженість роботи з web-запитами, відсутність можливості захисту на сервісному рівні та захисту доменних об'єктів, виникає актуальна потреба у використанні стороннього альтернативного рішення.

Для конфігурування безпеки доступу з урахуванням особливостей предметної області будуть використані елементи простору імен Spring Security, які доцільно подати такими групами функцій для відображення основних можливостей:

- Web/HTTP Security – забезпечує встановлення фільтрів, пов'язаних із сервісними компонентами, захист методів доступу до мережевих ресурсів, роботу аутентифікаційних механізмів програмної платформи, відображення сторінок помилок, аутентифікайції тощо;
- BusinessObject (Method) Security – відповідає за захист усіх рівнів сервісів;
- AuthenticationManager – здійснює обробку аутентифікаційних запитів від внутрішніх компонентів програмної платформи;
- AccessDecisionManager – забезпечує засоби доступу до web-сторінок та методів;
- AuthenticationProvider – забезпечує роботу менеджера аутентифікації користувачів та управління ідентифікаторами користувачів;
- UserDetailsService – використовуються програмною платформою як репозиторій користувачів, а також як основа стратегії доступу для AuthenticationProvider.

Під час реалізації модуля безпеки системи доцільно розглянути стандартний сценарій аутентифікації користувача і дослідити переваги, які для цього процесу надає Spring Security, з метою ефективного їх застосування. Стандартний сценарій складається з таких кроків: користувач намагається увійти у систему з певними даними для авторизації; система перевіряє коректність введених даних та відповідність паролю імені користувача; у разі успішної перевірки отримують контекст з інформацією про користувача, його роль тощо; цей контекст встановлюється для користувача; на основі встановленого контексту механізм контролю доступу надає користувачу можливість здійснювати визначений перелік дозволених операцій. Spring Security визначає такі корективи для стандартного механізму аутентифікації: отримані ім'я користувача та пароль комбінуються в екземпляр класу UsernamePasswordAuthenticationToken; цей об'єкт передається AuthenticationManager для проведення валідації; у разі успішності цієї процедури контекст безпеки встановлюється викликом методу SecurityContextHolder.getContext().setAuthentication(…) та передається всередині об'єкта аутентифікації [11].

Також важливо зазначити деякі функції управління сесіями:

- визначення перевищення ліміту відведеного часу підключення для однієї сесії та перехід на потрібний ресурс;
- управління паралельним доступом для одного й того самого користувача, що дає можливість налаштувати кожному користувачу можливість такого доступу;
- автоматичний захист від атаки "виправлення сесії", коли зловмисником створюються сесії, а потім відбуваються маніпуляції для входу легального користувача під цією сесією з метою перехоплення посилання, що містить ідентифікатор сесії та інші дані.

Spring Security підтримує функції роботи з паролями та зберігання їх у зашифрованому вигляді з використанням алгоритму хешування, що являє собою односторонню функцію перетворення деяких вихідних даних фіксованої довжини з вхідних даних довільної довжини. Зашифровані паролі зберігаються у базі даних системи, що мінімізує можливість їх викриття. Зламати подібний шифрований пароль можна лише повним перебором усіх можливих варіантів. Для збільшення надійності шифрування паролів Spring Security надає можливість використовувати додатковий рядок унікальних даних для кожного окремого користувача під час виконання алгоритму шифрування, що значно збільшує час, необхідний для розшифрування такого паролю.

Висновок

У результаті проведених досліджень та аналізу програмних платформ для розроблення модуля безпеки для web-орієнтованої системи підтримки прийняття рішень була обрана програмна платформа Spring Security. Вона надає можливість реалізації засобів безпеки за рахунок: підтримки різноманітних протоколів авторизації та їх налаштування; використання різних видів вбудованих фільтрів, що забезпечать функціонування модуля безпеки системи. До переваг Spring Security потрібно зарахувати можливість спочатку здійснювати налаштування функціональності модуля безпеки, а потім розширювати конфігурацію для досягнення необхідного рівня контролю над системою. Крім того, варто зазначити наявність детальної документації, постійний розвиток та удосконалення цієї платформи.

Проте необхідно чітко розуміти, що повноцінний захист системи неможливо реалізувати лише за допомогою використання бібліотек для авторизації користувачів або для запобігання мережевим атакам. Безпека системи дуже залежить від її архітектури, організації взаємодії модулів між собою та із зовнішніми системами.

1. Олійник Г. В., Грибков С. В., Литвинов В. А. Задача планування виконання договорів та підходи до її ефективного вирішення / Г. В. Олійник, С. В. Грибков, В. А. Литвинов // Математические машины и системы. – К. : ІПММС НАНУ, 2015. – С. 61–70. 2. Емельянова Н. З., Партыка Т. Л., Попов И. И. Защита информации в персональном компьютере. – М.: Форум, 2009. – 368 с. 3. Якименко І. З. Критерії оцінки рівня захисту комп'ютерних мереж з врахуванням їх архітектури // Інформатика та математичні методи в моделюванні. – 2013. – Т. 3, № 1 – С. 82–90. 4. Кононова В. О., Грибков С. В., Харкянен О. В. Оцінка засобів захисту інформаційних ресурсів / В. О. Кононова, С. В. Грибков, О. В. Харкянен // Вісник Національного університету "Львівська політехніка". – 2014. – № 806. – С. 99–105. 5. Шелупанова А. А., Груздева С. Л., Нахаева Ю. С. Аутентификация. Теория и практика обеспечения доступа к информационным ресурсам / А. А. Шелупанова, С. Л. Груздева, Ю. С. Нахаева. – М. : Горячая линия-Телеком, 2009. – 552 с. 6. Павленко Е. П. Разработка модуля защиты информационной системы полиграфического предприятия / Е. П. Павленко, И. А. Криворотенко, В. А. Айвазов // Вестник Нац. техн. ун-та "ХПИ" : сб. науч. тр. – Темат. вып.: Новые решения в современных технологиях. – Харьков : НТУ "ХПИ" 2013. – № 26 (999). – С. 30–34. 7. Apache Shiro Documentation [Електронний ресурс]. – режим доступа: http://shiro.apache.org/documentation.html, 11.03.2013. 8. Schaefer C., Ho C., Harrop R. Pro Spring 4th ed. Edition / Chris Schaefer, Clarence Ho, Rob Harrop. – Apress, 2014. – 728 p. 9. Mularien, P. Spring Security / P. Mularien. – PACKT Publications, 2010. – 396 с. 10. Уоллс К. Spring в действии / К. Уоллс. – М.: ДМК Пресс, 2013. – 752 с. 11. Ben Alex , Luke Taylor, Rob Winch, Gunnar Hillert Spring Security Reference [Електронный ресурс]. – режим доступа: http://docs.spring.io/spring-security/site/docs/4.1.3.RELEASE/reference/htmlsingle/, 2015.