УДК 004.056.5(079.2)

РОЗРОБКА ПОЛІТИКИ БЕЗПЕКИ *WEB*-ДОДАТКА ЕЛЕКТРОННОЇ КОМЕРЦІЇ ВІДПОВІДНО ДО ВИМОГ PA-DSS

Чолишкіна О. Г., канд. техн. наук, ***Гончар Г. Ю.***

Національний авіаційний університет

e-mail:grygoriy.gonchar@gmail.com

Проаналізовано вимоги стандарту PA-DSS. Визначено основні загрози web-додатків електронної комерції. Сформовано модель імовірного порушника. Розроблено політику безпеки для web-додатка, що виконує фунції оплати за допомогою кредитних карток, та побудованого за технологіями, суміжними з мовою програмування Java. Обґрунтовано вибір програмних засобів, що будуть використовуватись у ході реалізації захищеного web-додатка.

Ключові слова: *web*-додаток, PA-DSS, політика безпеки, вразливість, загроза.

The analysis of the requirements of the standard PA-DSS is provided. The main threats of e-commerce web-applications are analyzed. The security policy for Java-related web-appplication that performs payment functions by credit cards is formed. The choice of software to be used in the implementation of secure web-application is decided.

Keywords: *web*-application, PA-DSS, the security policy, vulnerability, threat.

Постановка проблеми

Електронна комерція як сфера економіки має тенденцію до швидкого зростання в Україні та світі. Упродовж останніх п’яти років український ринок електронної комерції показує щонайменший приріст на рівні 50—60 % незалежно від перманентних економічних коливань. Це відбувається завдяки перевагам електронної комерції як сфери бізнесу, а саме значному скороченню витрат, швидкому виведенню товару на ринок, низькій вартості розповсюдження цифрових продуктів та ін. Головною перевагою використання електронної комерції вважається масштабованість. До недоліків застосування такого бізнесу можна віднести сучасний стан безпеки електронних додатків. Якщо раніше увагу зловмисників було спрямовано на атаки мережевого рівня, то сьогодні на проблеми безпеки *web*-додатків, які призначені для здійснення покупок або оплати послуг через Інтернет. Переважна більшість таких додатків працює з матеріалом кредитних карток. При цьому персональні дані, які необхідні для оплати, вводяться у *web*-додатка та передаються глобальною мережею. Це може призвести до їх втрат.

Згідно зі статистичними даними, що надає компанія *Positive Technologies*, на 2010 р. майже половина *web*-додатків мають вразливості. Імовірність знайти критичну помилку у *web*-додатку за допомогою автоматичного сканування становить 35 та 80 % у результаті експертного аналізу. Це свідчить про те, що сучасні *web*-додатки вразливі не лише від атак кваліфікованих зловмисників, а навіть від атак новачків.

Стандарт *Payment Applications Data Security Standart* (PA-DSS) призначений для захисту *web*-додатків від загроз, що призводять до кражі даних кредитних карт. 1-го липня 2010 р. вступив у дію п’ятий мандат *Visa* щодо стандарту PA-DSS. Таким чином, проблема відповідності стандарту PA-DSS стає актуальною для всіх компаній, що мають намір обрати у своєму *web*-додатку як оплату кредитні картки.

Аналіз досліджень

PA-DSS — міжнародний стандарт в області інформаційної безпеки, створений *Payment Card Industry Security Standards Council* (PCI SSC), ґрунтується на стандарті *Payment Application Best Practices* (PABP). PA-DSS був створений для компаній, які розробляють програмні додатки у сфері електронної комерції, з метою забезпечення допомоги у проектуванні систем, що відповідають більш глобальному стандарту PCI DSS [1].

Область дії стандарту PA-DSS:

1. Програмні додатки, що розраховані для широкого вжитку без значних змін при впровадженні та налагодженні.

2. Програмні додатки, які переважно складаються з модулів, що можуть вибірково включатися залежно від вимог споживача та потрібних йому функцій. Отже, відповідати стандарту PA-DSS може лише модуль, який виконує платіжну функцію.

3. PA-DSS не поширюється на програмні додатки, що розроблені лише для одного користувача.

PA-DSS складається з 14 вимог у трьох напрямках: безпека програмного додатка, процес розробки, «Інструкція з впровадження».

Остання версія стандарту PA-DSS 2.0 була опублікована 1 жовтня 2010 р. та вступила в дію з 1 січня 2011 р. [2].

Стандарт PA-DSS містить вимоги і рекомендації щодо реалізації, уповадження і підтримки *web*-додатка, але серед них немає жодної, яка б вказувала, які саме програмні засоби для цього використовувати. Немає також ніякого іншого стандарту, який би вирішував цю проблему.

Таким чином, ***метою*** *даної статті* є розробка політики безпеки *web*-додатка електронної комерції за допомогою стандарту PA-DSS.

Для створення політики безпеки *web*-додатка найбільшу увагу приділемо саме вибору програмних засобів захисту та їх обґрунтуванню. Задля отримання можливості подальшого практичного застосування розробленої політики безпеки та розробленого *web*-додатка, необхідно, щоб *web*-додаток відповідав таким вимогам:

- можливість задовольнити вимоги стандарту PA-DSS;
- легкість упровадження (максимально можлива незалежність від платформи та середовища, в якому буде впроваджено *web*-додаток);
- розповсюдженість використаних технологій (наявність достатньої кількості навчальних матеріалів для спеціалістів на ринку праці);
- легкість у використанні (інтуїтивно зрозумілий інтерфейс та програмний код, який відповідає відомим шаблонам та практикам програмування);
- масштабованість (можливість розширити функціональність *web*-додатка без значної модифікації існуючого програмного коду та без значних витрат часу);
- продуктивність.

Об'єктом дослідження є захист *web*-додатка, що оброблює персональні дані кредитних карток. Оскільки стандарту PA-DSS може відповідати лише модуль, який виконує платіжну функцію, то як *предмет дослідження* буде використано *web*-додаток, що моделює роботу платіжного шлюзу (ПШ), який має модель преавторизація/виконання для *CNP*-трансакцій та розроблений за допомогою технологій Java EE 6, Spring Framework 3, Hibernate ORM framework, HTML, CSS, JavaScript та використовує базу даних Oracle Database 10 g.

ПШ являє собою програмно-апаратний комплекс, що дозволяє автоматизувати процес прийому платежів через Інтернет.

Будь-який ПШ належить до автоматизованих систем (АС) класу «3» та має підвищені вимоги до конфіденційності.

Розглянемо АС, що оброблює наступну інформацію.

Головний номер рахунку (PAN) — 16...19 — значний номер, який викарбований або закодований у пластикову картку та який ідентифікує рахунок власника картки:

– 1 цифра PAN — ідентифікатор індустрії пластикової картки (наприклад, для банківської та фінансової сфери 4 та 5);

– 1—6 цифри PAN — ідентифікатор емітенту (VISA: 4xxxxx, MasterCard: 51xxxx-55xxxx);

– 7— (N—1) цифри PAN — номер акаунту власника картки. Згідно стандарту безпеки PA-DSS v2, вимога 2.2 захисту потребують 7 — (N—4) цифри;

– N цифра — символ перевірки контрольної суми за алгоритмом Луна.

Ім'я та прізвище власника картки (*Cardholder Name*) — надруковано на картці великими латинськими літерами.

Сервіс-код (*Service Code*) — код відповіді платіжного додатка на запит користувача.

Кінцевий термін придатності картки, місяць та рік (*Expiration date*) також надруковані на картці.

Інформація магнітної стрічки (*Full Magnetic Stripe Data*) — дані з магнітної стрічки (в загальному випадку там зберігається CVC1 (*card validation code, MasterCard*) чи CVV1(*card verification value, Visa*) — код безпеки, необхідний для того, щоб переконатись, що картка була створена відповідним банком-емітентом.

Код безпеки CVC2/CVV2/CID — 3-х або 4-х значний код, потрібний для захисту віддалених транзакцій через Інтернет, надрукований на картці.

«CID» (*card identification number*) — код карток Discover.

«CVC» (*card validation code*) — код карток *MasterCard*.

«CVV» (*card verification value*) — код карток *Visa*.

PIN/PIN Block — PIN код (ідентифікаційний номер) картки.

Відповідно до стандарту PA-DSS, інформація, що оброблюється *web*-додатком, має правові режими, що зазначені в табл. 1.

Таблиця 1

Правові режими оброблюваної інформації

Правовий режим	Інформація
Перші 6 та останні 4 цифри — конфіденційна, решта цифр — таємна	PAN
Конфіденційна	Cardholder Name
Конфіденційна	Service Code
Конфіденційна	Expiration date
Таємна	CVC2/CVV2/CID

Виклад основного матеріалу

Як уже було зазначено, інформація, що обробляється *web*-додатком та потребує захисту — персональні дані кредитних карток. Тому політика безпеки має бути спрямована проти порушників, що намагаються несанкціоновано отримати доступ до цих даних.

Відповідно до нормативного документу [7] будемо розглядати модель порушника, що відповідає таким характеристикам. Мета: отримання конфіденційної й таємної інформації, що оброблюється в АС. Порушник має перший рівень доступу до АС. За рівнем знань про АС: володіння високим рівнем знань у галузі обчислювальної техніки та програмування, проектування та експлуатації АС. За методами і способами: використання винятково агентурні методи одержання відомостей. За місцем здійснення дії: без одержання доступу на контрольовану територію організації (АС).

Для формування моделі загроз будемо використовувати документ OWASP Top 10 *Application Security Risks* — документ, в якому перераховано 10 найбільш розповсюджених атак та уразливостей *web*-додатків та безліч посилань, спрямованих на допомогу в їх усуненні. OWASP Top 10 *Application Security Risks*, що був опублікований у 2010 р., включає такі вразливості та види атак [3]:

A1 — *Injection* (such as SQL, OS, and LDAP injection) — атака програмного додатка шляхом введення SQL запиту, команд операційної системи, LDAP-команд, спрямованих на ураження самої ОС, бази данних, *web*-серверу, інформації, що зберігається тощо, в доступні користувачу елементи *web*-додатка.

A2 — *Cross-Site Scripting* (XSS) — атака реалізована шляхом передачі на сторону клієнта програмного коду зловмисника (JavaScript, ActiveX, Java, Flash), спрямованого на порушення конфіденційності інформації користувача або порушення цілісності інформації у *web*-додатку.

A3 — *Broken Authentication and Session Management. Session Management* (Session Fixation) — перехоплення або генерація ID HTTP сесії авторизованого користувача. *Broken Authentication* (*Insufficient Authentication*) — вразливість, яка дозволяє порушнику здійснити несанкціонований доступ до ресурсів *web*-додатка, що потребують аутентифікації.

A4 — *Insecure Direct Object References* — вразливість, спричинена безпосереднім посиланням на об'єкт за його ім'ям.

A5 — *Cross-Site Request Forgery* (CSRF) — атака, яка змушує користувача надсилати *http*-запит на сторонній ресурс. Таким чином можливе перехоплення конфіденційної інформації користувача або застосування іншого виду атаки, наприклад, XSS.

A6 — *Security Misconfiguration* — вразливість, спричинена некоректним налаштуванням прав доступу користувачів *web*-додатка, бази даних, серверу тощо (часто є результатом того, що програмний код, який використовувався для тестування, не був вилучений).

A7 — *Insecure Cryptographic Storage* — вразливість, що виникає в результаті використання алгоритмів недостатньої криптостійкості або доступності зловмиснику виклику функцій дешифрування з інтерфейсу *web*-додатка.

A8 — *Failure to Restrict URL Access* — вразливість, що дозволяє здійснити доступ ресурсів *web*-додатка в обхід аутентифікації та авторизації.

A9 — *Insufficient Transport Layer Protection* — вразливість, спричинена недостатнім захистом інформації на транспортному рівні передачі даних. Незахищений мережевий трафік може бути перехоплений за допомогою спеціальних програм «сніферів».

A10 — *UnvalidatedRedirects and Forwards* — вразливість, яка дає змогу зловмиснику перенаправити користувача на задане ним посилання.

Повний перелік існуючих на сьогодні загроз *web*-додатків доступний у документі WASC *Threat Classification*[4].

Розглянемо побудову політики безпеки відповідно до стандарту PA-DSS 2.0. Порядок зберігання даних кредитних карток буде розроблено згідно з вимогою 1: не зберігати інформацію з магнітної стрічки, CAV2, CID, CVC2, CVV2 або PIN.

Відповідно до вимоги 1.1.1 *web*-додаток буде зберігати лише PAN, Cardholder Name, Service Code, Expiration date. Ці дані будуть зберігатись у таблиці логів платіжних транзакцій. Згідно з вимогою 1.1.2 CVC2/CVV2 після авторизації зберігатись не буде. Згідно з вимогою 1.1.5 програмний додаток після її реалізації має пройти додаткову перевірку на предмет того, що блок CVC2/CVV2 ніде не залишається після авторизації.

Згідно з вимогою 2.3 PAN буде зберігатись таким чином:

- цифри 1–6 — атрибут «1» таблиці бази даних;
- цифри 7–13 — атрибут «2» таблиці бази даних;
- цифри 13–17 — атрибут «3» таблиці бази даних.

До атрибутів 1.3 буде застосовано алгоритм AES-196 шифрування даних, реалізований засобами СУБД Oracle (*Transparent data encyption*).

Атрибут 2 буде зберігатись лише в результаті хеш-функції, реалізованої за алгоритмом SHA-1 засобами СУБД Oracle (пакет DBMS_CRYPTO).

Згідно з вимогою 3.1.1 кожен користувач АС буде мати унікальний ID.

Згідно з вимогою 3.1.2 аутентифікація буде проходити з використанням паролю.

Згідно з вимогою 3.1.3 пароль буде лише індивідуальним.

Згідно з вимогою 3.1.4 максимальний термін дії паролю 90 діб.

Згідно з вимогою 3.1.5 мінімальна довжина паролю 7 символів.

Згідно з вимогою 3.1.6 пароль буде складатись принаймні з літер та цифр.

Згідно з вимогою 3.1.7 програмний додаток буде зберігати історію паролів та вимагати, щоб паролі не повторювались.

Якщо акаунт було заблоковано, то згідно з вимогою 3.1.9 акаунт може розблокувати лише Адміністратор.

Згідно з вимогою 3.1.10 максимальна тривалість сесії буде 15 хв, після цього програмний додаток знову вимагатиме аутентифі-кацію

Згідно з вимогою 3.3 для зберігання паролів буде використовуватись лише криптоалгоритми, що використовують перевірені стандарти. Як перевірені стандарти будуть використані стандарти, що рекомендує корпорація Oracle та які реалізовані в СУБД Oracle Database 10g.

Згідно з вимогою 4.1 буде вестись аудит усіх спроб авторизації до *web*-додатка

Згідно з вимогою 4.2.1 буде вестись аудит усіх спроб доступу до інформації кредитних карток.

Згідно з вимогою 4.2.2 буде вестись аудит усіх дій з привілеями Адміністратора.

Згідно з вимогою 4.2.3 буде вестись аудит доступу до аудиту.

Згідно з вимогою 4.2.4 буде вестись аудит усіх невдалих спроб доступу в АС.

Згідно з вимогою 4.2.5 буде вестись аудит ідентифікації та аутентифікації.

Згідно з вимогою 4.2.6 буде вестись аудит ініціалізації системи аудиту.

Згідно з вимогою 4.3 програмний додаток у процесі аудиту буде зберігати ID користувача, тип події, дату та час, результат (успіх чи невдача), джерело події, ідентифікатор системного компонента або ресурсу.

Згідно з вимогою 4.3 аудит буде відбуватись централізовано та зберігатись у СУБД Oracle.

Згідно з вимогою 5.1 справжні кредитні картки не будуть використовуватись у ході тестування. Згідно з вимогою 5.1 буде використано OWASP Top 10 *Application Security Risks* як найкраща світова практика у захисті *web*-додатків.

Розглянемо перелік основних загроз *web*-додатка. У табл. 2 наведено основні типи загроз (що описано вище) та методи захисту їх від них.

Таблиця 2

Методи захисту *web*-додатка від основних видів загроз

Тип загрози	Перелік методів
A1	Використання фільтру ESAPI Encoder API, програмного інтерфейсу, який перевіряє вхідні дані на наявність SQL, OS, LDAP та інших команд та декодує їх у разі знаходження. Використання *Hibernate's prepared statements*. Перевага використання PL/SQL процедур. Надання лише найменш необхідних повноважень користувачам
A2	Фільтрування вхідних даних шляхом застосування ESAPI Encoder AP
A3	Ефективне управління механізмом аутентифікації, авторизації, управління сесіями та доступом до ресурсів. Реалізуємо за допомогою *Spring Security Framework*
A4	Використання архітектури *Model-View-Controller* (MVC), яка запобігає необхідності прямого посилання на об'єкти. Використання *Spring Security Access Control List*, який розмежовує права доступу користувачів до ресурсів *web*-додатка. Використання лише непрямих посилань, які залежать від ID http-сесії або від ID користувача
A5	Вбудова у посилання унікального ключа, що базується на ID користувача або ID сесії за допомогою OWASP's CSRF Guard
A6	Наявність чіткої архітектури *web*-додатка (MVC). Розмежування модулів *web*-додатка Конфігурації безпеки у *Spring Security*. Мінімізація прав доступу користувачів

Закінчення табл. 2

Тип загрози	Перелік методів
A7	Реалізація шифрування лише на стороні серверу засобами СУБД Oracle. Мінімізація засобів дешифрування, доступних через інтерфейс *web*-додатка. Використання лише найбільш крипостійких алгоритмів
A8	Використання засобів авторизації та аутентифікації Spring *Security* для всіх ресурсів *web*-додатка, доступ до яких обмежено. Використання *Spring Security* ACL для запобігання неавторизованного доступу
A9	Використання протоколу TLS (*Transport Layer Security*)
A10	URL посилання має формуватися не залежно від вхідних даних, у випадку, якщо це реалізувати неможливо, то потрібно перевіряти посилання на правильність та автентичність для відповідного користувача

Вимога 6. Бездротове з'єднання має бути захищеним.

Вимога 7. *Web*-додаток має бути протестований для знаходження вразливостей.

Тестування *web*-додатка на знаходження вразливостей буде виконано за допомогою *Acunetix Web Vulnerability Scanner* 7.0.

Вимога 8. Забезпечення захисту мережі.

Вимога 9. Дані кредитних карт не повинні зберігатись на сервері, що має доступ до мережі Інтернет.

Вимога 10. Має бути забезпечений захищений віддалений доступ.

Вимога 11. Конфіденційна інформація, що передається загальнодоступною мережею, повинна бути захищена криптографічними засобами захисту.

Вимога 12. Будь-який адміністративний доступ не через консоль має бути захищеним криптографічними засобами.

Вимога 13. Мають бути розроблені інструкції, тренувальні програми для споживачів та впроваджувачів програмного продукту.

Результатом є набір програмних рішень, що дасть змогу захистити *web*-додаток, і включає:

- *Spring Security Framework*;
- OWASP's CSRF *Guard*;
- ESAPI *Encoder* API;
- протокол TLS;
- засоби захисту та шифрування СУБД *Oracle* 10 g.

Наведемо опис та обґрунтування зазначених програмних засобів.

Технологія Java EE (Servlets, JSP)

Java — об'єктно-орієнтована мова програмування, розроблена компанією *Sun Microsystems*. Для розробки *web*-додатка була обрана ця мова програмування завдяки таким перевагам.

Програмні додатки на мові програмування *Java* можуть працювати на будь-якій віртуальній *Java*-машині (JVM) незалежно від комп'ютерної архітектури.

Засобів розробки та впровадження *Java* існує велика кількість, що забезпечує легкість розробки і впровадження *web*-додатка.

Java EE (скорочено від *Java Platform, Enterprise Edition*) — набір специфікацій та відповідної документації для мови програмування Java, в якому описано архітектуру серверної платформи для задач середніх та великих підприємств. *Java* EE є промисловою технологією і в більшості випадків використовується для високопродуктивних проектів, в яких необхідна надійність, масштабованість, гнучкість.

Отже, використання *Java* EE забезпечить високу продуктивність, масштабованість, легкість впровадження та розробки *web*-додатка.

СУБД *Oracle Database*

Має такі переваги.

Продукти *Oracle* інтегровані з технологіями мови програмування *Java*, тому вибір бази даних *Oracle* є виправданим. Серед інших переваг бази даних *Oracle*, що вплинули на вибір, можна виділити такі:

Технологія RAC (*Real Application Cluster*) — дозволяє об'єднати сервери баз даних у кластер. Технологія RAT (*Real Application Testing*) — полягає в тому, що база даних може відтворювати навантаження на базу даних з точністю із запланованою.

Active Data Guard — дає змогу створити резервний сервер бази даних, на якому відображувались би всі зміни основного серверу.

Oracle Database Firewall — програмний засіб, що проводить аудит активності бази даних у мережі.

Oracle Database Vault — програмний засіб, що контролює доступ до бази даних за такими параметрами, як час, IP адреса, назва програмного додатка, метод авторизації.

Oracle Label Security — програмний засіб, що класифікує дані за рівнем доступу та організовує доступ залежно від цих рівнів.

Oracle Advanced security — набір технологій, що дозволяють організувати прозоре шифрування даних як окремих атрибутів, так і таблиць та файлів даних у цілому.

У ході реалізації *web*-додатка будуть використані можливості *Oracle Advanced security*, а саме технології *Oracle Transparent Data Encryption* та пакет DBMS_CRYPTO.

Для шифрування в Oracle 10*g* R2 доступні алгоритми 3DES168, AES128, AES192. Як було зазначено раніше, в питаннях надійності алгоритмів будемо покладатись на рекомендації Oracle, то як алгоритм для шифрування буде використано AES192.

Для реалізації хеш-фунцій буде використано пакет DBMS_CRYPTO, який надає такі алгоритми для шифрування: MD5, SHA-1, MD4. Серед запропонованих алгоритмів будемо використовувати SHA-1, який вважається більш криптостійким.

Бібліотеки для ефективного програмування на *Java — Spring, Hibernate*

Spring framework являє собою джерело розширень, потрібних для ефективної розробки складних бізнес-додатків без важких програмних моделей. *Spring framework* обраний для збільшення ефективності процесу проектування та розробки *web*-додатка, а також завдяки наявності дочірнього набору розширень *Spring Security*.

Hibernate — бібліотека мови програмування Java, призначена для розв'язання задач об'єктно-реляційного проектування (*object-relational mapping* — ORM). *Hibernate* звільняє розробника від значного обсягу низькорівневого програмування SQL та JDBC коду шляхом автоматичного генерування запитів до бази даних. Це зменшує час розробки *web*-додатка, полегшує перехід від однієї СУБД до іншої, а отже, полегшує впровадження *web*-додатка.

Spring Security

Spring Security — Java / Java EE бібліотека, що надає механізми побудови систем аутентифікації та авторизації, а також інші можливості забезпечення безпеки для промислових додатків, створених за допомогою *Spring Framework*. *Spring Security* має такі можливості, що повинні бути використані в ході розробки *web*-додатка.

Захист URL доступу. Захист URL доступу організуємо шляхом застосування сервлетів-фільтрів до *http*-запитів. Налагодження безпеки відбувається в елементі <html> XML-файлу.

Авторизація

Серед існуючих варіантів виберемо авторизацію через *web*-форму. На створеній формі можна розмістити всю необхідну інформацію про послуги *web*-додатка. Таким чином, можна уникнути створення окремих сторінок з інформацією про послуги *web*-додатка, які потребують створення ролі ROLE_ANONYMUS.

У *web*-додатку будуть створені такі ролі користувачів:

ROLE_USER — для користувачів та операторів *web*-додатка;

ROLE_ADMIN — для адміністраторів *web*-додатка.

Розподілення доступу відбувається на основі аутентифікації користувачів, характеристик інформаційних ресурсів та атрибутів доступу. У *Spring Security* розподілення доступу виконується менеджерами доступу — класами, які реалізують інтерфейс *AccessDecisionManager*.

У *web*-додатку буде реалізовано такий набір конфігурацій задля забезпечення максимальної безпеки та захисту від загрози A8.

Менеджер розподілення доступу *Consensus-Based* з такими воутерами: *RoleVouter*, *IpAdressVoter* (воутер реалізується самостійно). Оскільки у *web*-додатку відсутній анонімний (неавторизований доступ), сервіс *remember-me*, то потреби в *AuthenticationVoter* немає.

Spring Security дає можливість розподілити доступ не тільки до URL, а й до методів, усередині сервлетів. Серед існуючих засобів оберемо розподіл прав доступу до методів за допомогою анотацій.

Програмний інтерфейс ESAPI Encoder API, OWASP CSRF Guard

ESAPI (The OWASP *Enterprise Security* API) — *open source* бібліотека для вільного користування, що містить засоби поліпшення безпеки *web*-додатків. OWASP ESAPI for Java EE версії 2.0 підтримує Java 1.5. У *web*-додатку будемо користуватись декодером користувацького введення для захисту від SQL Injections (ESAPI.*encoder*(). encodeForSQL (*stringVariable*)) та від XSS. Оскільки єдине місце *web*-сторінки, куди можуть бути імпортовані користувацькі дані — це всередину *html* тегів, потрібно скористуватись фільтром ESAPI.*encoder*().encodeForHTML(commentary). Дані фільтри доцільніше реалізувати на рівні *getters* та *setters* методів POJO-класів.

Протокол захищеної передачі даних TLS (*Transport Layer Security*) — криптографічний протокол, що забезпечує захищений обмін даними між вузлами в мережі Інтернет. Протокол заснований на протоколі Netscape SSL версії 3.0 та складається з двох частин — *TLS Record Protocol* та TLS *Handshake Protocol*.

Висновки

Розроблено комплекс методів для створення політики безпеки *web*-додатка електронної комерції, побудованого на базі мови програмування *Java* та бази даних *Oracle*. Результуюча політика безпеки має використовуватись у ході розробки *web*-додатка, що імітує роботу модуля оплати за допомогою кредитних карток, а також інших *web*-додатків, що використовують вказані технології. *Web*-додаток, побудований за даною політикою, доцільно застосовувати як модуль оплати для Інтернет-магазину або іншого *web*-додатка, що потребує модуль оплати за допомогою кредитних карток.

Для захисту *web*-додатків електронної комерції, особливо додатків, які забезпечують функції оплати, необхідно використовувати стандарт PA-DSS. Приступаючи до розробки *web*-додатка, необхідно ознайомитись з провідними документами в галузі захисту *web*-додатків.

Для захисту *web*-додатка необхідно користуватись існуючими бібліотеками та програмними інтерфейсами безпеки, розробленими провідними IT організаціями.

Відповідно до отриманої в ході дослідження політики безпеки, як фільтр вхідної інформації *web*-додатка доцільно використовувати бібліотеку OWASP ESAPI. Для *web*-додатків *Java*, що використовують бібліотеку *Spring*, обов'язково слід використовувати можливості *Spring Security*.

Під час вибору бази даних ефективним є використання *Oracle Database*, яка забезпечує повний набір засобів, необхідних для відповідності вимогам PA-DSS. Обов'язковим з погляду безпеки є перевага реалізації бізнес-логіки на стороні серверу, а не на стороні клієнта. Для реалізації захищеної передачі даних через Інтернет доцільно використовувати протокол TLS.

Серед питань, що можуть бути розглянуті в подальших дослідженнях, слід виділи порядок упровадження та підтримки захищеного *web*-додатку. Це питання великою мірою залежить від IT інфраструктури організації, де буде впроваджено *web*-додаток і має значну кількість різноманітних варіацій.

ЛІТЕРАТУРА

1. *Payment* Card Industry Security Standards Council standarts. — https://www.pcisecuritystandards.org/security_standards/documents.php.

2. *Payment* Card Industry Security Standards Council. Payment Application Data Security Standard (PA-DSS) 2.0, 2010. — 55 p.

3. *The Open* Web Application Security Project (OWASP). The OWASP Top 10 Web Application Security Risks for 2010. — https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

4. *WASC* Threat Classification. — http://wasc.org/doc/WASC_Threat_Classification

5. *Ron Ben* Natan. HOWTO Secure and Audit Oracle 10g and 11g. — CRC Press, 2009. — 454 p.

6. *Gary Mak*, Josh Long, Daniel Rubio. Spring Recipes. — Apress, 2010. — 1059 p.

7. *Peter Mularien*. Spring Security 3. — Publishing, 2010. — 397 p.

8. *НД ТЗІ* 1.6-003-2004. Модель загроз для інформації та модель порушника. — 23 с.

Стаття надійшла до редакції 10.04.11.