DOI: 10.18372/2410-7840.20.13423
УДК 004.4:004.75:004.455(043.2)

МАСШТАБУВАННЯ ТА ПОСИЛЕННЯ ЗАХИСТУ ДАНИХ ВЕБ-ДОДАТКУ ВІДПОВІДНО ДО ВИМОГ СТАНДАРТІВ PCI DSS, HIPAA/HITECH, FEDRAMP

Михайло Колпаков, Андрій Петренко

Збільшення кількості інформаційних веб-ресурсів зумовлено розвитком технологій взаємодії з інформацією, доступністю та простотою у використанні для споживача, можливостями автоматизації бізнес процесів, заощадження ресурсів і часу на надання послуг для підприємців. Зі зростанням обсягу даних, що обробляються комп'ютеризованими системами, проблема гнучкості архітектури масштабування та посилення захисту даних набула значної актуальності. В даній статті розглянуто комплексний підхід, щодо впровадження механізмів вирішення вищезазначених проблем відповідно до вимог стандартів PCI DSS, HIPAA/HITECH, FEDRAMP. Запропоновано ефективний підхід, щодо інтеграції Amazon S3 хмарного сховища у веб-додатки, написані мовою програмування Java. Покрокова практична інструкція впровадження веб-сервісів дозволить не лише ефективно масштабувати та захистити дані продукту, а також значно розширити функціональні можливості веб-додатку, використовувати внутрішні аналітичні інструменти для моніторингу активності користувачів, генерувати звіти на основі зібраної статистики. Розглянуто математична модель алгоритму шифрування AES, що реалізований в хмарному середовищі Amazon, для пояснення доцільності та актуальності його використання. Побудовано графік порівняння швидкості найактуальніших алгоритмів шифрування на основі даних, отриманих шляхом проведення дослідження з заміру часу витраченого на шифрування, при різних об'ємах даних. Даний підхід дозволить програмним продуктам відповідати вимогам визначених директивою з захисту даних ЄС та FISMA, покращити масштабування даних за рахунок зон та регіонів доступності, посилити захист даних за рахунок внутрішніх механізмів Amazon, таких як: контроль доступу, аудит, мережевий брандмауер, шифрування на стороні серверу та інфраструктура управління ключами шифрування.

Ключові слова: *хмарне сховище, масштабування даних, шифрування, Amazon S3, PCI DSS, HIPAA/HITECH, FEDRAMP..*

Вступ

Проблема масштабованості та захисту даних є досить актуальною для сучасних підприємств у зв'язку з глобальним переходом на цифровий формат інформації, розвитком електронного документообігу та автоматизацією обробки даних, що надає значні переваги в напрямку оптимізації бізнес процесів та заощадженні ресурсів та часу на надання послуг.

Зі збільшенням обсягу даних бізнес-аналітики до яких може отримувати доступ через інтернет постійно зростаюча аудиторія постачальників, виробників продукції та послуг основну увагу починають приділяти масштабованості. Масштабованість – здатність працювати з додатковими користувачами або транзакціями шляхом нарощування ресурсів без фундаментальної перебудови архітектури або моделі реалізації [1]. Одним з найбільш ефективних та гнучких підходів є побудова серверної інфраструктури на базі регіонів та зон доступності і взаємодія між клієнтом та сервером на базі REST API архітектурного підходу.

Безпека даних є найактуальнішою проблемою інформаційних систем на сьогодні. Щороку кількість атак на інформаційні ресурси зростає в рази, а тому програмні рішення, щодо захисту об'єктів інформаційної системи користуються значним попитом. Оцінка ефективності захисту інформаційно-комунікаційної системи відбувається на основі критеріїв, що описані в стандартах інформаційної безпеки ISO/IEC та визнані міжнародною спільнотою. В даній статті ми розглянемо найактуальніші стандарти безпеки даних різних сфер діяльності, а саме PCI DSS – стандарт безпеки даних індустрії платіжних карт [2], HIPAA/HITECH – вимоги до обробки і зберігання медичної приватної інформації [3], FedRAMP – вимоги захищеності процесу доступу до хмарних продуктів та сервісів [4].

Аналіз існуючих досліджень

У роботі [5] проведено загальний огляд Amazon S3 сховища та функціональних можливостей програмного продукту, а праця [6] містить аналіз комплекту розробки програмного забезпечення Amazon SDK. Процес проходження сертифікації PCI DSS описаний в роботі [7], HIPAA/HITECH – в праці [8]. Проте комплексного рішення починаючи від інтеграції хмарного сховища в власні програмні продукти до проходження сертифікації відповідності міжнародним вимогам захисту даних відповідно до директив ЄС та FISMA на даний момент відсутня у науковій літературі. Таким чином, виникають труднощі при розробці чи впровадженні хмарного сховища до

власних веб-додатків, що відповідатимуть вищеперерахованим вимогам.

Метою даної роботи є модифікація веб-додатку шляхом інтеграції Amazon S3 хмарного сховища для підвищення ефективності масштабування та захисту даних відповідно до стандартів PCI DSS, HIPAA/HITECH, FedRAMP та директив захисту даних ЄС і FISMA.

Основна частина дослідження

Масштабування

Основним способом масштабування баз даних є підхід розділення даних на групи та виділення їх на окремі сервери. На базі даного підходу застосовують дві основні стратегії: реплікація і шардінг.

Реплікація дозволяє створити повний дублікат бази даних, так замість одного серверу використовують декілька. Найчастіше використовується схема розподілу даних master-slave. Master – це основний сервер БД, куди надходять усі дані. Всі зміни в даних (додавання, оновлення, видалення) повинні відбуватися на цьому сервері. Slave – це допоміжний сервер БД, який копіює всі дані з майстра. З цього серверу зчитують дані. Таких серверів може бути декілька. Операція читання даних відбувається набагато більше ніж запитів на зміну даних. Тому реплікація дозволяє зменшити навантаження на основний сервер за рахунок переносу операцій читання на slave сервер. Слід зазначити, що реплікація не дуже зручний механізм масштабування. Причиною тому є розсинхронізація і затримки, а в копіюванні даних з майстра на слейв. Проте це відмінний засіб для забезпечення відмовостійкості. Завжди існує можливість переключитися на слейв, якщо майстер вийде з ладу і навпаки. Найчастіше реплікація використовується спільно з шардінгом саме з міркувань надійності.

Суть шардінгу полягає в поділі бази даних на окремі частини так, щоб кожну з них можна було винести на окремий сервер. Цей процес залежить від структури бази даних і виконується прямо в додатку на відміну від реплікації [9].

Хмарне сховище Amazon S3 використовує підхід масштабування даних на базі регіонів та зон доступності (рис. 1). Кожен регіон AWS включає в себе кілька окремих, фізично ізольованих зон доступності, з'єднаних між собою мережами з низькою затримкою і високою пропускною здатністю. Відмовостійкість можна підвищити за рахунок реплікації даних між різними географічними регіонами. Це можна зробити як за допомогою приватних високошвидкісних мережних підключень, так і за допомогою публічних інтернет-з'єднань, забезпечуючи таким чином додатковий рівень стабільності бізнесу або доступ з низькими затримками у міжнародному масштабі.

Підсумовуючи вищезазначений матеріал слід акцентувати увагу на покращенні ефективності, швидкості роботи та надійності збереження даних базуючись на підході регіонів та зон доступності. Перш за все – це значне зменшення часу затримки операцій з даними, шляхом визначення місцезнаходження користувача системи та підбором найоптимальніших центрів обробки даних. Користувачі повністю контролюють регіон розміщення, в якому дані розташовані фізично, і доступ до них, що дозволяє легко забезпечити відповідність регіональним нормативним вимогам і вимогам до розміщення даних.

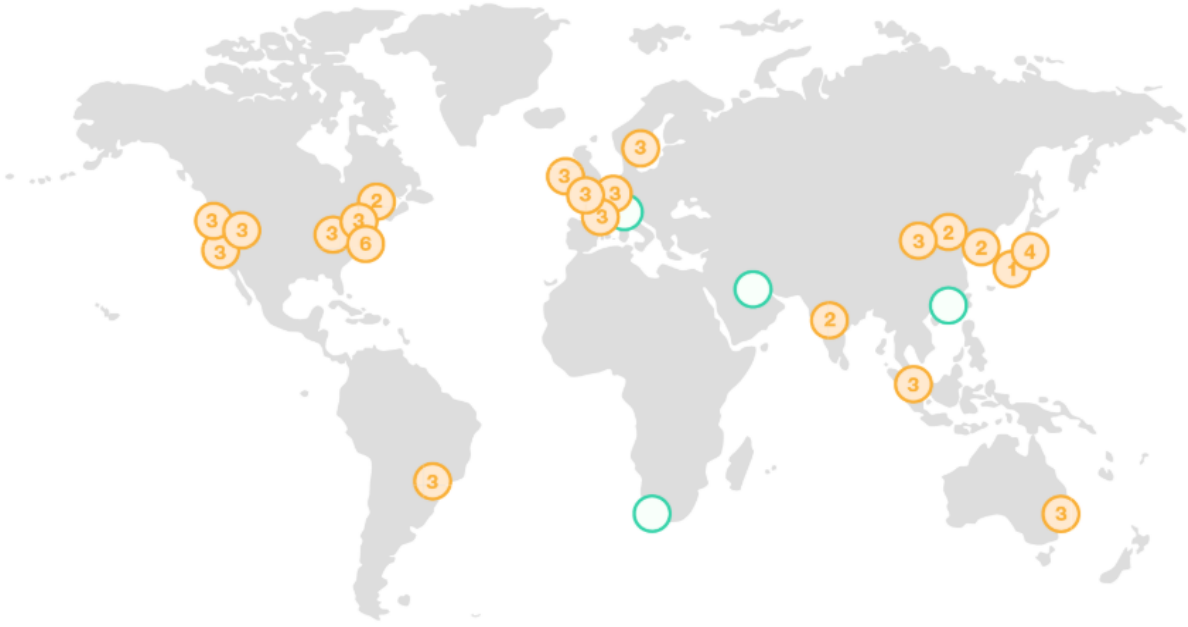

Рис. 1. Регіони та зони доступності центрів обробки даних Amazon Simple Storage Service

Захист даних

На основі існуючих загроз порушення конфіденційності, доступності та цілісності даних виділяють наступні напрямки захисту: шифрування, контроль доступу, аудит подій і брандмауери. Amazon S3 надає наступні інструменти в кожному з підходів:

– Шифрування даних при передачі для всіх сервісів за допомогою протоколу TLS. Гнучкі варіанти управління ключами, що дозволяють клієнтам вибирати, чи зберігати повний контроль над ключами шифрування або передати управління ключами AWS. Черги зашифрованих повідомлень для передачі конфіденційних даних за допомогою шифрування на стороні сервера. Виділені апаратні сховища криптографічних ключів дозволяють клієнту забезпечити відповідність нормативним вимогам.

– Мережеві брандмауери, вбудовані в Amazon і можливості брандмауера інтернет-додатків сервісу AWS WAF дозволяють створювати приватні мережі і керувати доступом до інформаційних об'єктів і додатків.

– Повна інформація про виклики API, включаючи відомості про те, хто, звідки, коли і який виклик зробив. Можливості об'єднання журналів, автоматизації їх аналізу та створення звітів про відповідність нормативним вимогам. Сповіщення про настання певних подій чи перевищенні порогових значень.

– Можливості безпечного управління доступом до сервісів та ресурсів AWS. Підтримка багатофакторної автентифікації, включаючи варіанти апаратної автентифікації. Інтеграція і федералізація з корпоративними директоріями для зменшення адміністративних накладних витрат і поліпшення умов роботи кінцевих користувачів.

Перелічені механізми захисту даних дозволяють веб-додаткам, що інтегрували хмарне сховище Amazon S3 у власні програмні продукти відповідати вимогам нормативних документів з інформаційної безпеки Європейського Союзу та директив захисту даних FISMA.

Збережені дані можуть бути зашифровані на стороні сервера використовуючи один із найефективніших алгоритмів блочного шифрування AES256. Алгоритм AES складається з наступних етапів:

1. Дані розбиваються на блоки по 128 біт.
2. Використовується поле GF(2^8) – це числа 0..255, для котрих визначені операції множення та додавання.

2.1 Візьмемо число з поля та представимо у вигляді восьми біт: $a = a_7a_6a_5a_4a_3a_2a_1a_0$.

2.2 Аналогічно візьмемо число b.

3. Додавання a + b = a xor b.
4. Для множення запишемо многочлени за коефіцієнтами бітів цих чисел:

$$p = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0;$$

$$q = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0.$$

5. На останньому етапі перемножимо ці многочлени та знайдемо остачу від ділення на m, що по суті є зашифрованим байтом інформації:

$$m = x_8 + x_4 + x_3 + x + 1;$$

$$r = p \times q \bmod (m).$$

Відповідно до дослідження швидкості шифрування блочних алгоритмів AES, DES та 3DES (рис. 2) можна зробити висновок, що використання алгоритму AES є досить затратним з точки зору ресурсів на обчислення. В той же час з точки зору надійності та криптостійкості AES значною мірою переважає своїх конкурентів.

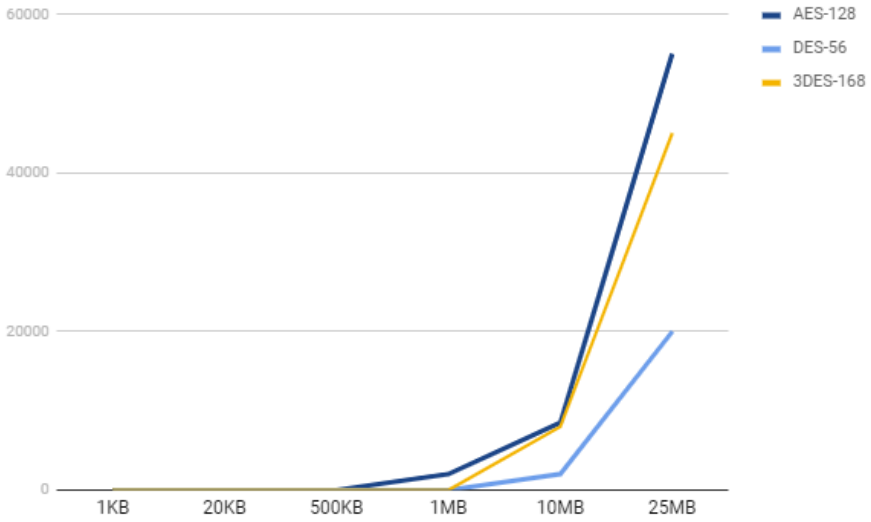

Рис. 2. Порівняльний графік швидкості обчислення алгоритмів AES, DES та 3DES

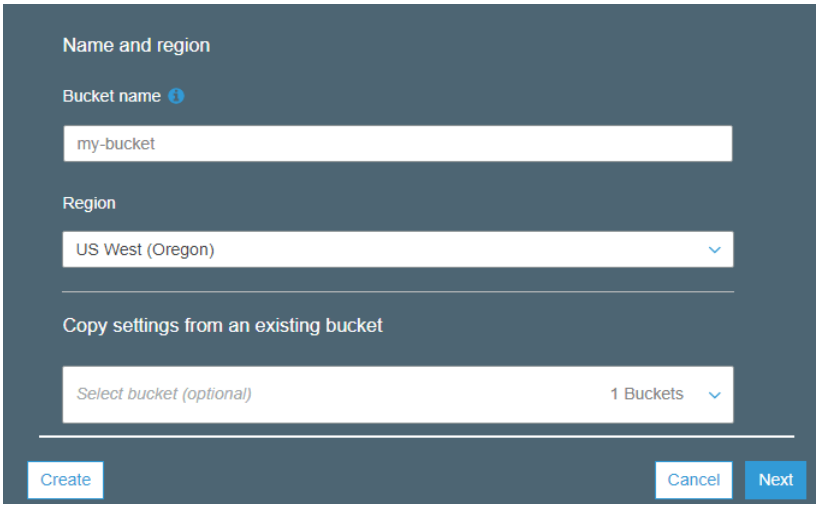

Рис. 3. Створення контейнеру для даних в Amazon S3 сховищі

Далі необхідно впровадити залежність у конфігураційний файл проекту для підключення Amazon SDK. У нашому випадку інтегруємо сховище в проект на базі засобу автоматизації та складання програм написаних мовою програмування Java – Maven. Для цього опишемо залежність до конфігураційного файлу pom.xml (рис. 4).

```
<dependency>
   <groupId>com.amazonaws</groupId>
   <artifactId>aws-java-sdk</artifactId>
   <version>1.11.133</version>
</dependency>
```

Рис. 4. Програмний код підключення Amazon SDK

Останнім кроком буде написання REST API контролерів, за унікальними ідентифікаторами яких будуть виконуватися операції завантаження та видалення файлу до віддаленого сховища, використовуючи класи раніше підключеної бібліотеки Amazon SDK та Spring MVC фреймворк для імплементації контролерів [10].

Спочатку створимо клас BucketController котрий використовує анотацію @RestController та @RequestMapping для вказання компілятору, що даний клас є контролером до якого можна звернутись по URL "/storage" (рис. 5). Далі необхідно впровадити залежність AmazonClient класу всередину контролера, для цього скористуємося анотацією @Autowired над конструктором. Останнім етапом є створення методів загрузки та видалення файлів. Використовуючи анотацію @Post Mapping та @DeleteMapping вкажемо ідентифікатор та HTTP метод котрий повинен використовувати клієнт при запиті до відповідного URI.

Скористаємося середовищем розробки Postman, що дозволяє протестувати API відсилаючи запит по обраному URI, з можливостю налаштування заголовків і тіла запиту та метод HTTP запиту. Введемо ідентифікатор ресурсу у спеціально відведену строку, виберемо метод POST, файл з локальної файлової системи, який ми хочемо віддалено зберегти та виконаємо запит. В результаті отримаємо 200 код відповіді, що означає успішний запит. Перейдемо до створеного на попередньому етапі контейнеру файлів та впевнимося в тому, що файл збережений (рис. 6).

Отже процес інтеграції Amazon S3 сховища у власні програмні продукти є досить простим та не потребує значних часових та ресурсних затрат. Дані, що зберігаються в хмарному середовищі можуть бути оброблені внутрішніми аналітичними інструментами, репліковані в межах одного або декількох регіонів, зашифровані одним із найстійкіших алгоритмів блочного шифрування AES 256.

```
@RestController
@RequestMapping("/storage/")
public class BucketController {

    private AmazonClient amazonClient;

    @Autowired
    BucketController(AmazonClient amazonClient) {
        this.amazonClient = amazonClient;
    }

    @PostMapping("/uploadFile")
    public String uploadFile(@RequestPart(value = "file")
MultipartFile file) {
        return this.amazonClient.uploadFile(file);
    }

    @DeleteMapping("/deleteFile")
    public String deleteFile(@RequestPart(value = "url") String
fileUrl) {
        return this.amazonClient.deleteFileFromS3Bucket(fileUrl);
    }
}
```

Рис. 5. Програмний код REST контролера для завантаження та видалення даних

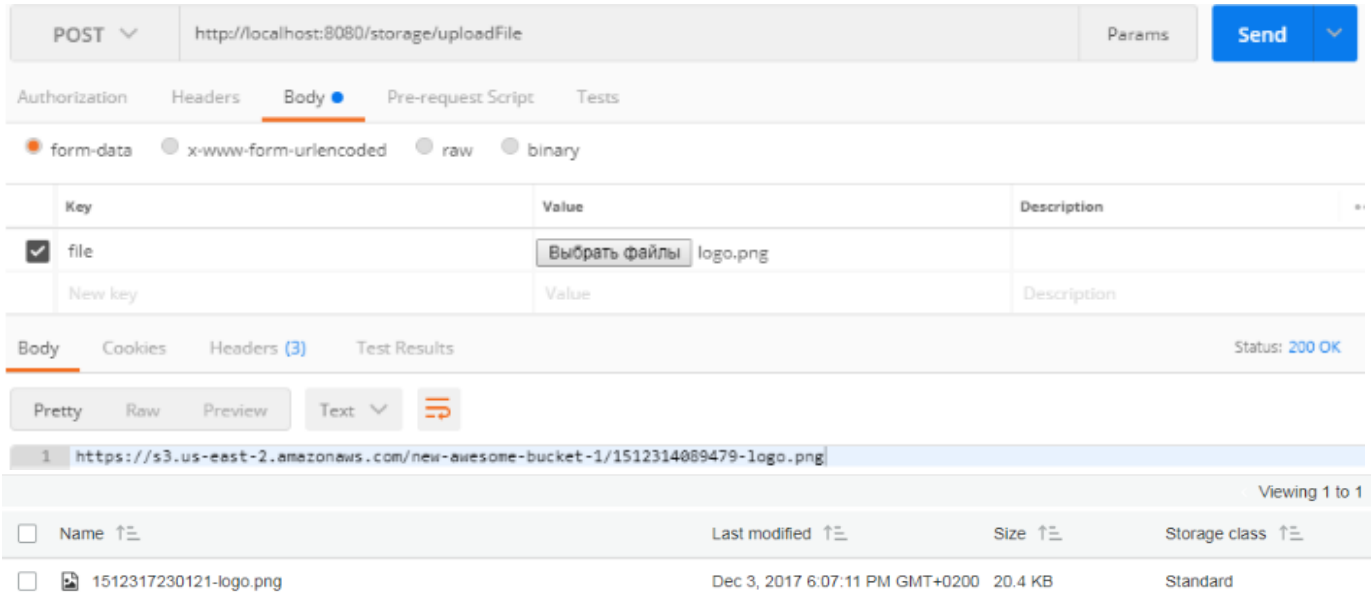

Рис. 6. Приклад завантаження файлу

Висновки

В даній статті запропоновано модифікацію програмного забезпечення для використання об'єктного хмарного сховища Amazon S3, що дозволяє взаємодіяти з внутрішніми хмарними сервісами через REST API інтерфейс. Оскільки Amazon використовує масштабування на базі регіонів та зон доступності модифікований додаток підвищує швидкість передачі даних. Модифікований додаток, в частині перенесення даних до Amazon S3 хмарного сховища, отримує ще один рівень захисту даних і відповідає вимогам стандартів захисту інформації PCI DSS, HIPAA/HITECH, FEDRAMP та директив захисту даних ЄС і FISMA.

Наукова новизна роботи полягає у розробці нового комплексного підходу для впровадження ефективного механізму масштабування та посилення захисту даних веб-додатків, шляхом інтеграції Amazon S3 хмарних сховищ. Даний підхід дозволить ефективніше вирішити проблему масштабування даних, за рахунок фізичної наявності обчислювальних машин в різних географічних регіонах та взаємодії користувача з найближчим серед них, що скорочує час стандартних операцій зчитування, додавання та видалення даних. Оскільки об'єкти веб-сервісу захищені набором інструментів, таких як: між мережевий брандмауер, шифрування на стороні серверу, аудит, багатофакторна автентифікація та контроль доступу – це дозволить автоматично використовувати перелічені інструменти для захисту даних веб-додатків та відповідати вимогам міжнародних стандартів захисту інформації PCI DSS, HIPAA/HITECH, FEDRAMP та директив захисту даних ЄС і FISMA.

ЛІТЕРАТУРА

[1]. Інформаційно-навчальний портал програмування EASYCODE. [Електронний ресурс]. Режим доступу: http:// easy-code. com. ua/ 2011/ 01/ masshtabovanist-sistem-biznes-analitiki.

[2]. Payment Card Industry Security Standards глобальний форум для визначення стандартів безпеки в сфері електронних платежів. [Електронний ресурс]. Режим доступу: https://www.pcisecuritystandards.org.

[3]. K. Brian, Fox, Daniel M. Atchinson, The Politics Of The Health Insurance Portability And Accountability Act, *Health Affairs*, pp. 146-150.

[4]. Державний сайт з визначення стандартів безпеки для хмарних продуктів та сервісів FedRAMP. [Електронний ресурс]. Режим доступу: https://www.fedramp.gov.

[5]. Науковий журнал «Хакер». [Електронний ресурс]. Режим доступу: https://xakep.ru/2010/08/12/52949.

[6]. Статті компанії з розробки програмного забезпечення Oril. [Електронний ресурс]. Режим доступу: https://medium.com/oril.

[7]. Блог компанії Payonline. [Електронний ресурс]. Режим доступу: https://habr.com/company/payonline/blog/130652.

[8]. Кібербезпека в охороні здоров'я [Електронний ресурс]. Режим доступу: https://www.trendmicro.com/ru_ru/business/capabilities/solutions-for/healthcare.html.

[9]. Інформаційний портал в сфері масштабування та оптимізації інформаційних ресурсів [Електронний ресурс]. Режим доступу: https://ruhighload.com.

[10]. Комп'ютерна інженерія і кібербезпека: досягнення та інновації: матеріали Всеукр. наук.-практ. конф. здобувачів вищої освіти й молодих учених (м. Кропивницький, 27-29 листоп. 2018 р.), М-во освіти і науки України, Держ. наук. установа "Інститут модернізації змісту освіти", Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2018, 448 с.

МАСШТАБИРОВАНИЕ И УСИЛЕНИЕ ЗАЩИТЫ ДАННЫХ ВЕБ-ПРИЛОЖЕНИЯ В СООТВЕТСТВИИ С ТРЕБОВАНИЯМИ СТАНДАРТОВ PCI DSS, HIPAA / HITECH, FEDRAMP

Увеличение количества информационных веб-ресурсов обусловлено развитием технологий взаимодействия с информацией, доступностью и простотой в использовании для потребителя, возможностями автоматизации бизнес-процессов, экономией ресурсов и времени на оказание услуг для предпринимателей. С ростом объема обрабатываемых данных компьютеризированными системами, проблема гибкости архитектуры масштабирования и усиления защиты данных приобрела значительную актуальность. В данной статье рассмотрен комплексный подход, по внедрению механизмов решения вышеупомянутых проблем в соответствии с требованиями стандартов PCI DSS, HIPAA/HITECH, FEDRAMP. Предложен эффективный подход, по интеграции Amazon S3 облачного хранилища в веб-приложения, написанные на языке программирования Java. Пошаговая практическая инструкция внедрение веб-сервисов позволит не только эффективно масштабировать и защитить данные продукта, а также значительно расширить функциональные возможности веб-приложения, использовать внутренние аналитические инструменты для мониторинга активности пользователей, генерировать отчеты на основе собранной статистики. Рассмотрена математическая модель алгоритма шифрования AES, который реализован в облачной среде Amazon, для объяснения целесообразности и актуальности его использования. Построен график сравнения скорости актуальных алгоритмов шифрования на основе данных, полученных путем проведения исследования по замеру времени, затраченного на шифрование при различных объемах данных. Данный подход позволит программным продуктам соответствовать требованиям определенными директивой по защите данных ЕС и FISMA, улучшить масштабирование данных за счет зон и регионов доступности, усилить защиту данных за счет внутренних механизмов Amazon, таких как: контроль доступа, аудит, сетевой брандмауэр, шифрование на стороне сервера и инфраструктура управления ключами шифрования.

Ключевые слова: облачное хранилище, масштабирование данных, шифрование, Amazon S3, PCI DSS, HIPAA / HITECH, FEDRAMP.

SCALING AND ENHANCING DATA PROTECTION FOR WEB APPLICATION DATA IN ACCORDANCE WITH PCI DSS, HIPAA/HITECH, FEDRAMP STANDARDS

Annotation. The increase in the number of information web resources is due to the development of technologies for interaction with information, accessibility and ease of use for the consumer, opportunities for automating business processes, saving resources and time for providing services for entrepreneurs. In this paper, an integrated approach is considered regarding the implementation of the mechanisms for solving the above problems in accordance with the requirements of the standards of PCI DSS, HIPAA / HITECH, FEDRAMP. An effective approach for integrating the Amazon S3 cloud storage into web applications written in the Java programming language is proposed. The step-by-step guide to web services implementation will not only effectively scale and protect product data, but also significantly extend the functionality of the web application, use internal analytical tools to monitor user activity, generate reports based on aggregated statistics. The step-by-step guide for integration of web services will not only allow effectively to scale and protect product data, but also significantly expand the functionality of the web application, to use internal analytical tools for monitoring user activity, to generate reports based on collected statistics. The mathematical model of the AES encryption algorithm, implemented in the Amazon cloud environment, is considered to explain the feasibility and relevance of its use. A graph of comparing the speed of the most current encryption algorithms based on the data obtained by conducting research on measuring the amount of time spent on encryption at different data volumes was constructed. This approach will allow software products to meet the requirements of the EU and FISMA data protection directives, improve data scaling by accessibility zones and regions, and enhance data protection through internal Amazon mechanisms such as access control, auditing, network firewall, server-side encryption and the encryption key management infrastructure.

Keywords: cloud storage, data scaling, encryption, Amazon S3, PCI DSS, HIPAA/HITECH, FEDRAMP.

Колпаков Михайло Олексійович, студент кафедри комп'ютеризованих систем захисту інформації Навчально-наукового інституту комп'ютерних інформаційних технологій Національного авіаційного університету.
E-mail: mishanyakolpakov@gmail.com.
Колпаков Михаил Алексеевич, студент кафедры компьютеризированных систем защиты информации учебно-научного института компьютерных информационных технологий Национального авиационного университета.
Kolpakov Mykhailo, student of computer information protection systems Training and Research Institute of Computer Information Technologies of the National Aviation University.

Петренко Андрій Борисович, к.т.н., доцент, доцент кафедри комп'ютеризованих систем захисту інформації Навчально-наукового інституту комп'ютерних інформаційних технологій Національного авіаційного університету.
E-mail: pab.05@ukr.net.
Петренко Андрей Борисович, к.т.н., доцент, доцент кафедры компьютеризированных систем защиты информации учебно-научного института компьютерных информационных технологий Национального авиационного университета.
Petrenko Andriy, Ph.D., associate professor, assistant professor of computer information protection systems Training and Research Institute of Computer Information Technologies of the National Aviation University.