УДК 004.056.5

МОДЕЛЮВАННЯ ПРОЦЕДУРИ ЗАХИСТУ ВЕБ-РЕСУРСІВ ТА ВЕБ-СЕРВІСІВ

Гончарук Богдан[1], Арсенюк Ігор[1]

Вінницький національний технічний університет, Хмельницьке Шоссе 95

Анотація

Запропоновано та проаналізовано рішення для реалізації процедури захисту веб-сервісів та веб-служб із використанням додаткового ключа (гранта) юзера, для подальшого доступу до сервера авторизації і, як наслідок, отримання програмного токена доступу неопосередковано до приватних даних.

Abstract

An approach is proposed on implementing secure procedures for web-services with using additional key (grant) to receive access to authorization server and get access token which is necessary for using private data.

Вступ

У сучасному світі гігантськими темпами зростають обсяги різноманітної інформації. А як відомо, інформація є дуже дорогим продуктом. Звичайно, не будь-яка. Не дарма ж німецький банкір Натан Майер Ротшильд сказав свою крилату фразу: „Хто володіє інформацією – той володіє світом". Ми часто стикаємось з такими поняттями як „комерційна таємниця", „критична інформація", „інформаційна безпека", „персональні дані" тощо. За потрібну „цікаву" інформацію часто борються різноманітні конкуренти (фізичні особи, фірми, організації, компанії, банківські установи і навіть цілі країни). Нажаль, з часом, кількість людей, які хочуть заволодіти „цікавою" інформацією несанкціоновано, не зменшується, а зростає, шо, робить вкрай важливим і актуальним питання захисту такої інформації та обмеження доступу до неї. Безумовно, проблема захисту та обмеження доступу – це класична проблема, що буде актуальною допоки є інформація, яку треба захищатистити. Не дивно, що разом із еволюціонуванням інформаційних технологій та комп'ютерних потужностей, еволюціонують також інструменти і захисту, і атак на інформацію, як у мережі Інтернет, так і поза її межами. Нижче захист інформації розглядається у контексті Інтернет мережі.

Аналіз запропонованої процедури захисту

У даній роботі проаналізовано та модифіковано процедуру верифікації користувача із використанням коду авторизації та токена доступу, який, на відміну від багатьох відомих алгоритмів, містить додаткові дані, необхідні для отримання доступу, а саме – грант, який потрібен для "спілкування" з сервером, що генерує програмний токен [1 – 3].

Застосування додаткового гранта (ключа) юзера дає можливість підвищити анонімність та безпеку опрацювання запитів на кожній інстанції модуля. А завдяки додатковим алгоритмам шифрування, котрі можуть бути використані для додаткового опрацювання усіх ключів ми можемо забезпечити так звану "одноразовість" усіх даних, що застосовуються для ідентифікації, автентифікації та авторизації користувача, адже кожен ключ, токен і навіть сесія з'єднання будуть мати свій "термін придатності". По завершенню цього терміну будь-які згенеровані, у процесі, дані стануть неактуальними і модуль буде вимагати у користувача повторного введення даних для валідації (наприклад, адреса електронної пошти, номер телефону, пароль, додатковий SMS-код, у випадку, якщо увімкнено багатофакторну авторизацію).

Варто зазначити також і те, що такі заходи безпеки здатні зекономити багато коштів, але, тим не менш, зекономити на цьому, в основному, можуть компанії та підприємства у яких є достатньо ресурсів на підтримку та розробку такої архітектури, тому дане рішення варто розглядати у контексті співвідношення "ціна рішення / вартість даних".

Для захисту веб-додатку або веб-служби за допомогою вищевказаної процедури, слід реалізувати програмну інфраструктуру, що міститиме такі компоненти:

- Юзер-агентор (зацікавлена сторона або сторони), що володіє даними на захищеному веб-ресурсі.
- Юзер-агент – аплікація, за допомогою якої реалізується взаємодія клієнта та веб-додатку.
- Веб-додаток – ресурс, який здійснює первинну ідентифікацію, автентифікацію та авторизацію та надає доступ до запитуваної юзер-агентом інформації.
- Сервер авторизації – сервер, що містить два незалежні ендпоінти: для генерації, видачі короткотривалого програмного токена та здійснення авторизації, видачі авторизаційного коду.
- Ресурс сервер – це ресурс, що містить дані.

Проаналізуємо основний процес взаємодії юзера та веб-додатку:

- Під час первинного запиту юзера, веб-додаток перенаправляє користувача на сервер авторизації, що відповідає за генерацію відповідного інтерфейсу для первинної ідентифікації та авторизації користувача (рис. 1).
- Юзер вводить необхідні дані (логін, пароль, тощо) та надсилає на сервер авторизації.
- Сервер авторизації здійснює валідацію даних, генерує код авторизації та надсилає на юзер-агент інструкцію переадресації на веб-сервер із використанням коду авторизації.
- Юзер-агент надсилає код авторизації на веб-сервер.
- Веб-сервер надсилає код авторизації на сервер авторизації.
- Сервер авторизації здійснює валідацію, генерує короткотривалий програмний токен доступу та токен для поновлення сесії і надає його веб-серверу з якого був здійснений запит на авторизацію.
- Веб-сервер, у свою чергу, здійснює подальші запити до ресурс серверу із наданням отриманого програмного токена доступу.
- Ресурс-сервер здійснює валідацію токена доступу, якщо потрібно – дешифрує додаткові дані, проводить усі необхідні для перевірки достовірності агентора операції та повертає результат, що може містити потрібні дані або відмову з поясненням, у чому полягає проблема.

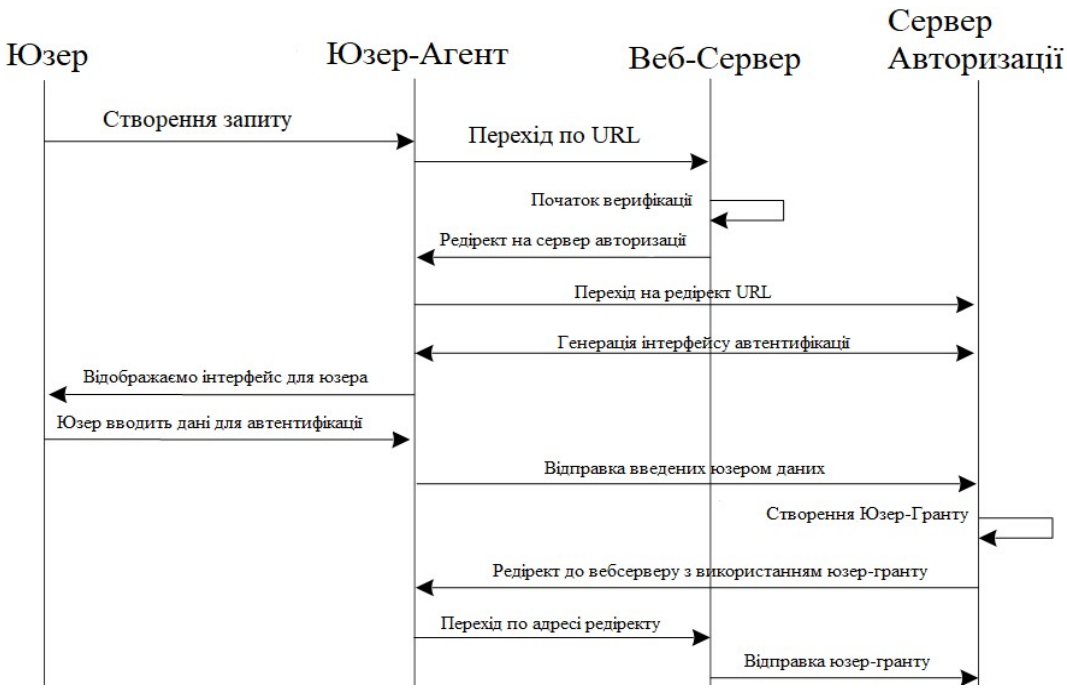

Рисунок 1 – Алгоритм роботи задіяних інстанцій захищеного додатку

На рис. 1 наведено різні ітерації, через які проходить запит користувача переш ніж у нього з’явиться доступ хоча б до якихось даних. При цьому, програмний токен, який надається у разі успішної автентифікації має короткий термін існування та перевіряється під час кожного запиту даних. Саме завдяки забезпеченню короткочасності програмного токену та декількох ітерацій верифікації валідності даних юзера, підвищується ступінь захисту від програм-інтерсепторів (сніферів), мережних-шпіонів, які можуть перехоплювати дані та надсилати їх зловмисникам.

Для підвищення надійності кожну інстанцію, яка розміщена в окремому виділеному мережному просторі, слід інкапсулювати та абстрагувати від інших таким чином, щоби вона мала доступ лише до тієї інформації, яка потрібна для виконання конкретних та чітко прописаних функцій [2].

Крім того, у разі правильної організації архітектурних компонентів, кожна з інстанцій фізично не буде мати ніякої інформації про інстанції на рівень абстракції вище наступної. Тобто, юзер-агент взагалі ніяк не зможе контактувати з сервером, що генерує програмні токени, оскільки він про нього нічого не знає. Таким чином архітектура має чітку обмеженість, і якщо представити кожну інстанцію у вигляді вершини графа – отримаємо орієнтований ациклічний граф.

Наочніше, взаємодію користувача з елементами веб-додатку можна зобразити як наведено на рис. 2:

Рисунок 2 – Процес взаємодії юзера та веб-додатку

Як бачимо, на наведених рисунках архітектурні блоки мають обмежений доступ лише до тих елементів архітектури, з якими вони взаємодіють. Тобто, мати доступ до однієї з інстанцій недостатньо, адже у разі збою, чи, наприклад, непередбачуваної (у даному разі непрописаної та не покритої тестами) поведінки одного з блоків, інші блоки можуть розірвати зв’язок та зупинити взаємодію. А це приведе до повного припинення процесу до початку наступної сесії авторизації, або мануального запуску після перевірки цілісності даних, сертифікатів та мережних зв’язків.

Одним із найголовніших недоліків даної організації архітектурних компонентів подібних систем захисту – висока ціна. Виділені сервери, налаштування та підтримка сертифікатів, маршрутизаторів, швидкості обробки запитів, метрики, системи підтримки “здоров’я” мережі – усе це потребує ретельного, чітко визначеного технічного опису, проектування, тестування і т. п. Саме тому такий підхід частіше використовується у разі підвищеної потреби у захисті даних, а також, коли компанія володіє достатнім бюджетом для планування та експлуатації такої системи [4].

Узагальнена схема взаємодії агентора з модулем захисту на макро-рівні наведена на рис. 3.

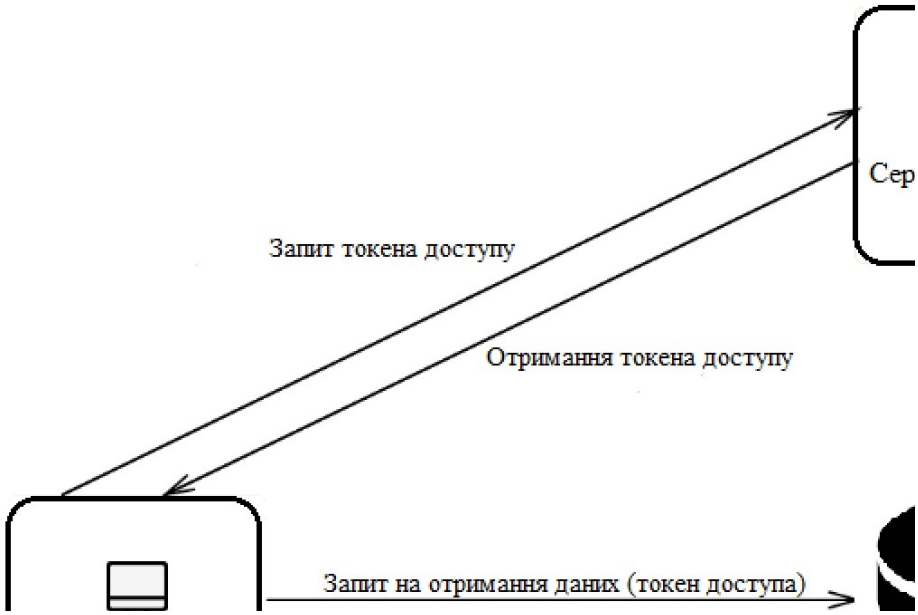

Рисунок 3 – Спрощена схема взаємодії з модулем захисту

Висновок

Запропоновано процедуру верифікації, аналізу та передачі персональних даних, з використанням гранта та програмного токена доступу, який дає можливість перевірити достовірність агентора на кожному етапі взаємодії. При цьому забезпечується багатоетапна перевірку та відстеження актуальності даних клієнта, що надсилає запити до веб-додатку. Дана процедура відрізняється від класичних сценаріїв авторизації тим, що містить додаткові ключі та обмеження тривалості сесії взаємодії агентора з модулем захисту.

Проаналізовано бізнес-складову запропонованого рішення, яке показує, що на його створення, підтримку, оновлення і т. п. потрібні інженери відповідного кваліфікаційного рівня та додаткове обладнання (виділені сервери, мережі серверів, cloud-технології), і, очевидно, захищена та детальна документація роботи кожної інстанції (в т. ч. алгоритмів шифрування, дешифрування, роботи мережних серверів, проксі, VPN, якщо вони використовуються у конкретній архітектурі).

Список використаних джерел:

1. OAuth [Електронний ресурс] – Режим доступу: https://docs.oracle.com/cd/E39820_01/doc.11121/gateway_docs/content/part_oauth.html

2. Microsoft identity platform and OAuth 2.0 authorization code flow – [Електронний ресурс] – Режим доступу: https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-auth-code-flow

3. Authorization code grant flow [Електронний ресурс] – Режим доступу: https://docs.axway.com/bundle/APIGateway_762_OAuthUserGuide_allOS_en_HTML5/page/Content/OAuthGuideTopics/oauth_flows_auth_code.htm

4. Гончарук Б. Г., Арсенюк І. Р. Захист веб-додатків та веб-служб за допомогою багатоетапної верифікації з використанням потоку авторизації через веб-сервер // Матеріали XLIX науково-технічної конференції підрозділів ВНТУ. Вінниця, 2020. URL: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2020/paper/view/8992/7757