МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____С.В. Казмірчук
«_____»_____20__ р.

На правах рукопису

УДК 004.056.5:510.22(043.3)

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема:Удосконалена система автентифікації та керування доступом до веб-додатків

Виконавець: А.В. Коляка

Науковий керівник: к.т.н. О.О. Висоцька

Нормоконтролер: к.т.н. О.О. Висоцька

Київ 2020

ВСТУП

Стрімкий розвиток обчислювальних мереж і особливо мережі Інтернет призвів до повсюдного використання веб-додатків різного призначення. У всьому світі активно йде процес децентралізації обчислень, що дозволяє створювати масштабовані системи, здатні обслуговувати величезну кількість користувачів з високою продуктивністю.

У зв'язку з широким розповсюдженням електронних ресурсів у всіх сферах людської діяльності гостро стоїть завдання забезпечення інформаційної безпеки в таких системах. Однією з основних заходів щодо захисту даних є забезпечення надійної автентифікації користувача. На даний момент існує багато підходів до автентифікації і ще більше реалізацій цих підходів.

При цьому не всі класичні рішення задачі автентифікації підходять для використання у всіх веб-додатках. А різні типи систем пред'являють свої унікальні вимоги до підсистем автентифікації. Крім того, активний розвиток обчислювальної техніки дозволяє легко зламувати алгоритми автентифікації, які ще 10-15 років тому вважалися надійними. У зв'язку з цим ведеться безперервна робота в області дослідження і розробки методів автентифікації. Дослідженнями методів автентифікації займалися такі вчені як: Альфонс Бертільйон, Ерік Гроссе, М. Болл, Дж. Х. Коннел та інші.

Актуальність роботи визначається тим, що проблеми пов’язані з розмежуванням доступу і автентифікацією користувачів є критичними і удосконалення механізмів вирішення цих задач необхідне для забезпечення коректної роботи сучасних веб-додатків.

Метою дипломної роботи є розробка удосконаленої системи автентифікації та керування доступом до веб-додатків, яка забезпечить надійну автентифікацію та авторизацію при доступі до веб-додатків за рахунок використання сучасних методів автентифікації та контролю доступу.

Для досягнення поставленої мети вирішуються наступні задачі:

- дослідити сучасні технології автентифікації та керування доступом;
- розробити алгоритми автентифікації користувачів різних типів до веб-додатку, а також алгоритми розмежування доступу;
- програмно реалізувати систему автентифікації та контролю доступу, яка забезпечує надійність, безпеку та зручність для її користувачів на основі розроблених алгоритмів.

Методи дослідження базуються на основі статистичного аналізу систем автентифікації та їх вразливостей (для розробки алгоритму автентифікації та контролю доступу) та об’єктно-орієнтованого програмування (для програмної реалізації розроблених алгоритмів).

Галузь застосування. Розроблена удосконалена система та запропоновані в ній механізми відносяться до галузі інформаційної безпеки і можуть бути використані для підвищення надійності автентифікації та авторизації при доступі до веб-додатків, за рахунок використання сучасних методів автентифікації та контролю доступу.

Об’єктом дослідження є процеси автентифікації та керування доступом до веб-додатків.

Предметом дослідження є технології, методи та системи двофакторної автентифікації та розмежування доступу користувачів до веб-додатків.

Наукова новизна одержаних результатів полягає в наступному: удосконалено процес автентифікації користувачів до веб-додатку, який забезпечує підвищений рівень захисту та водночас є зручним у користуванні, завдяки технології єдиного входу, додаткової функції перевірки при двофакторній автентифікації та вдосконаленому механізму перевірки прав користувачів.

Практична цінність отриманих результатів: полягає у створенні клієнт-серверного програмного рішення для реалізації процесів автентифікації та керування доступом користувачів у веб-додатках, що забезпечує

підвищений рівень захисту завдяки удосконаленому алгоритму автентифікації та перевірки прав користувачів і може використовуватись у веб-додатках з різними типами користувачів для розмежуванням прав доступу.

Апробація. Основні положення роботи буде розглянуто на XVI Міжнародній науково-практичній конференції: "Наука та освіта" 22-30 грудня 2020, Чехія.

РОЗДІЛ 1
ТЕХНОЛОГІЇ АВТЕНТИФІКАЦІЇ

1.1 Нормативно-правова база

Законність функціонування є однією з головних вимог до системи електронної автентифікації та контролю доступу. Ця законність повинна базуватися на сукупності правових документів і підзаконних нормативних актів, які спрямовані на створення необхідних умов для захисту національних інтересів в інформаційній та інших сферах життя країни. Найбільш значущими нормативними документами в галузі інформаційної безпеки, визначальними критеріями для оцінювання захищеності електронних веб-додатків і вимогами, пропонованими до механізмів захисту, є:

- Керівні документи «Загальні положення про технічний захист інформації в Україні», «Загальні положення з захисту інформації в комп'ютерних системах від НСД» та «Вимоги до засобів електронної ідентифікації, рівнів довіри до засобів електронної ідентифікації для їх використання у сфері електронного урядування» [18];
- Закон України «Про електронні довірчі послуги» [20];
- Загальні критерії оцінки безпеки IT (The Common Criteria for Information Technology Security Evaluation / ISO 15408) [19].

1.1.1 Положення про технічний захист інформації в Україні

Правові та організаційні засади технічного захисту важливої для держави, суспільства і особи інформації, охорона якої забезпечується державою відповідно до законодавства, визначає Положення про технічний

захист інформації в Україні, затверджене Указом Президента України від 27 вересня 1999 р. № 1229 (зі змінами та доповненнями).

Відповідно до п. 1 Положення, технічний захист інформації (ТЗІ) – діяльність, спрямована на забезпечення інженерно-технічними заходами конфіденційності, цілісності та доступності інформації щодо органів державної влади, органів місцевого самоврядування, органів управління Збройних сил України та інших військових формувань, утворених згідно з законодавством України, відповідних підприємств, установ, організацій (далі органів, щодо яких здійснюється ТЗІ). Основними завданнями органів, щодо яких здійснюється ТЗІ, є: забезпечення ТЗІ згідно з вимогами нормативно-правових актів з питань ТЗІ; видання у межах своїх повноважень нормативно-правових актів із зазначених питань; здійснення контролю за станом ТЗІ (п. 11 Положення) [18].

Правову основу ТЗІ в Україні становлять Конституція України (254к/96-ВР), закони України, законодавчі акти Президента України та Кабінету Міністрів України, нормативно-правові акти Служби безпеки України, Адміністрації Державної служби спеціального зв'язку та захисту інформації України, інших державних органів, міжнародні договори України, згода на обов'язковість яких надана Верховною Радою України, з питань ТЗІ, а також це Положення. Згідно з п. 8 Положення, суб'єктами системи ТЗІ є: Державна служба спеціального зв'язку та захисту інформації України (ДССЗЗІ України); органи, щодо яких здійснюється ТЗІ; науково-дослідні та науково-виробничі установи ДССЗЗІ України, державні підприємства, що перебувають в управлінні ДССЗЗІ України та виконують завдання з питань ТЗІ; військові частини, підприємства, установи та організації всіх форм власності й громадяни-підприємці, які провадять діяльність з ТЗІ за відповідними дозволами або ліцензіями; навчальні заклади з підготовки, перепідготовки та підвищення кваліфікації фахівців з ТЗІ. Державна політика ТЗІ формується згідно з законодавством і реалізується ДССЗЗІ України у взаємодії з органами,

щодо яких здійснюється ТЗІ. Нормативно-правові акти з ТЗІ є обов'язковими для виконання всіма суб'єктами системи ТЗІ [18].

Оцінювання захищеності інформації здійснюється шляхом атестації або експертизи комплексів ТЗІ та інспекційних перевірок. За результатами атестації або експертизи комплексів ТЗІ визначається можливість введення в експлуатацію об'єкта, де циркулюватиме інформація, охорона якої забезпечується державою [19].

1.1.2 Закони України про електронні довірчі документи

Під час створення сучасної та ефективної системи забезпечення ІБ істотного значення набуває наявність відповідної нормативно-правової бази, без якої неможливо охопити всі сфери життєдіяльності суспільства в рамках єдиного правового поля, розробити загальнонаціональну концепцію розвитку держави й ефективно реалізовувати політику національної безпеки в інформаційній сфері. Це означає, що всі без винятку дії щодо захисту й реалізації національних інтересів України в будь-якій сфері й на будь-якому рівні передусім спираються на чинне законодавство України [19].

Важливим для забезпечення ІБ України документом є Закон України «Про електронні довірчі документи» № 440-IX зі змінами від 14.01.2020 (далі – Закон). Закон визначає правові та організаційні засади надання електронних довірчих послуг, у тому числі транскордонних, права та обов’язки суб’єктів правових відносин у сфері електронних довірчих послуг, порядок здійснення державного нагляду (контролю) за дотриманням вимог законодавства у сфері електронних довірчих послуг, а також правові та організаційні засади здійснення електронної ідентифікації.

Метою цього Закону є врегулювання відносин у сферах надання електронних довірчих послуг та електронної ідентифікації. Згідно закону електронна довірча послуга - послуга, яка надається для забезпечення

електронної взаємодії двох або більше суб’єктів, які довіряють надавачу електронних довірчих послуг щодо надання такої послуги [20].

1.2 Автентифікація в кібербезпеці

Більшість сучасних систем захисту паролів для Інтернету є недосконалими. Користувачі відвідують веб-сайти, які мають форми реєстрації, які повідомляють наскільки захищений вибраний пароль. Інформація про це лунає у новинах, її можна помітити в електронних листах, які користувачі отримують кожні кілька місяців, нагадуючи їм про оновлення паролів.

Може скластися враження, що користувачі використовують тільки надійні та складні паролі для своїх облікових записів але в більшості випадків це не так. Цілком імовірно, що небезпечні паролі є одними з найбільших вразливостей, які існують у організації.

Автентифікація - це процес перевірки особистості користувача. Технологія перевірки автентичності забезпечує контроль доступу для систем, перевіряючи, чи облікові дані користувача збігаються з обліковими даними в базі даних авторизованих користувачів або на сервері автентифікації даних.

Користувачі, як правило, ідентифікуються з ідентифікатором користувача, а автентифікація виконується, коли користувач надає облікові дані, наприклад пароль, який відповідає цьому ідентифікатору користувача. Більшість користувачів найбільш знайомі з використанням пароля, який, як частина інформації, повинен бути відомий тільки користувачеві.

Після автентифікації користувач або процес, як правило, також піддаються процесу авторизації, щоб визначити, чи має доступ об'єкт до захищеного ресурсу або системи. Користувач може бути автентифікований, але не може отримати доступ до ресурсу, якщо користувачеві не було надано дозволу на доступ до нього. Терміни автентифікації та авторизації часто

використовуються як взаємозамінні. Хоча вони часто можуть бути реалізовані разом, ці дві функції відрізняються.

Хоча автентифікація є процесом перевірки ідентичності зареєстрованого користувача перед тим, як дозволити доступ до захищеного ресурсу, авторизація є процесом перевірки того, що авторизований користувач отримав дозвіл на доступ до запитаних ресурсів. Процес, за допомогою якого доступ до цих ресурсів обмежується певною кількістю користувачів, називається контролем доступу[1].

Автентифікація користувача відбувається в більшості взаємодій від людини до комп'ютера. Як правило, користувач повинен вибрати ім'я користувача або ідентифікатор користувача та надати дійсний пароль для початку використання системи.

Багато компаній використовують автентифікацію для перевірки користувачів, які входять до їхніх веб-сайтів. Без відповідних заходів безпеки дані про користувачів, такі як номери кредитних і дебетових карт, а також номери соціального страхування, можуть потрапити в руки кіберзлочинців.

Організації також використовують автентифікацію, щоб контролювати, які користувачі мають доступ до корпоративних мереж і ресурсів, а також визначати та контролювати, які процеси та сервери мають до них доступ. Компанії також використовують автентифікацію, щоб дозволити віддаленим працівникам безпечно отримувати доступ до своїх програм і мереж.

Для підприємств та інших великих організацій автентифікацію можна здійснити за допомогою системи єдиного входу (SSO), яка надає доступ до декількох систем з єдиним набором облікових даних для входу.

Рисунок 1.1 - Технологія SSO

Наприклад, у великих приватних мережах часто існує декілька незалежних підсистем. З допомогою SSO реалізується доступ до всіх підсистем без повторного введення логіну/паролю. Для цього користувачеві буде достатньо ввести логін/пароль лише раз для однієї з підсистем і він матиме доступ до всіх інших.

1.3 Як працює автентифікація

Під час автентифікації облікові дані, що надаються користувачем, порівнюються з даними в базі даних авторизованих користувачів або в локальній операційній системі або через сервер автентифікації. Якщо облікові дані збігаються, а авторизований об'єкт має право використовувати цей ресурс, процес завершується і користувачеві надається доступ. Повернення дозволів і папок визначають як середовище, яке бачить користувач, так і спосіб взаємодії з ним, включаючи години доступу та інші права, такі як обсяг місця для зберігання ресурсів.

Традиційно автентифікація здійснювалася за допомогою систем або ресурсів, до яких здійснювався доступ; наприклад, сервер автентифікує користувачів, використовуючи власну систему паролів, реалізовану локально,

використовуючи ідентифікатори для входу (імена користувачів) і паролі. Кожен користувач спочатку реєструється, використовуючи призначений або самозваний пароль. При кожному наступному використанні користувач повинен знати і використовувати попередньо оголошений пароль.

Проте протоколи додатків мережі, HTTP і HTTPS, не мають статусу, а це означає, що сувора автентифікація потребує повторної автентифікації кінцевих користувачів кожного разу, коли вони отримують доступ до ресурсу за допомогою HTTPS. Замість того, щоб навантажувати кінцевих користувачів цим процесом для кожної взаємодії через Інтернет, захищені системи часто покладаються на автентифікацію на основі маркерів, в якій автентифікація виконується один раз на початку сеансу. Система автентифікації видає підписаний маркер автентифікації до програми кінцевого користувача, і цей маркер додається до кожного запиту від клієнта [2].

Автентифікацію об'єктів для систем і процесів можна здійснювати за допомогою облікових даних, які працюють як ідентифікатор користувача та пароль, за винятком того, що дані облікового запису автоматично подаються відповідною програмою. Вони також можуть використовувати цифрові сертифікати, які були видані та перевірені центром сертифікації як частину інфраструктури відкритого ключа для автентифікації особи при обміні інформацією через Інтернет.

1.4 Ідентифікаційні фактори

Автентифікація користувача з ідентифікатором і паролем, зазвичай, є найбільш поширеною. Оскільки цей тип автентифікації спирається тільки на один фактор автентифікації, це тип однофакторної автентифікації.

Сильна автентифікація - це термін, який формально не був визначений, але зазвичай використовується, щоб визначити, що тип автентифікації, який використовується, є більш надійним і стійким до атак.

Фактор автентифікації являє собою деякий фрагмент даних або атрибут, який може використовуватися для автентифікації користувача, який запитує доступ до системи.

Наразі використовуються такі фактори автентифікації:

- Фактор знань: "Те, що ви знаєте";
- Фактор володіння: "Те, що у вас є";
- Фактор невідповідності: "Те, ким ти є";
- Розташування фактора: "Де ви знаходитесь";
- Фактор часу: "Час автентифікації".

Незважаючи на те, що використовуються в якості додаткових факторів автентифікації, розташування користувача і поточний час самі по собі не є достатніми, без принаймні одного з перших трьох факторів. Тим не менш, поширеність смартфонів допомагає полегшити процес багатофакторної автентифікації для багатьох користувачів. Більшість смартфонів обладнані системою GPS, що дозволяє впевнено підтверджувати місце входу в систему; MAC-адреси смартфонів також можуть використовуватися для автентифікації віддаленого користувача.

1.5 Традиційні методи

Традиційна автентифікація залежить від використання файлу паролів, в якому ідентифікатори користувачів зберігаються разом з хешами паролів, пов'язаних з кожним користувачем. Під час входу в систему пароль, що подається користувачем, змішується і порівнюється зі значенням у файлі паролів. Якщо два хеши збігаються, користувач автентифікується.

Сильна автентифікація зазвичай відноситься до автентифікації, яка використовує принаймні два фактори з різними типами. Відмінність важлива оскільки і ім'я користувача, і пароль можна вважати типом фактора знань, можна сказати, що базове ім'я користувача та автентифікація паролем

використовують два фактори знань для автентифікації - однак це не вважається формою двофакторної автентифікації (2FA).

Двофакторна автентифікація зазвичай залежить від фактора знання, що поєднується з біометричним фактором або фактором володіння, наприклад, маркером безпеки. Багатофакторна автентифікація може включати будь-який тип автентифікації, що залежить від двох або більше факторів, але процес автентифікації, який використовує пароль, плюс два різних типи біометричних даних, не вважатиметься трьохфакторною автентифікацією, хоча, якщо процес вимагає фактора знання, володіння і фактор невідповідності, її такої можна вважати [16].

Такий підхід до автентифікації має ряд недоліків, особливо для ресурсів, розгорнутих в різних системах. По-перше, зловмисники, які можуть отримати доступ до файлу паролів для системи, можуть використовувати атаки на хешовані паролі для вилучення паролів. З іншого боку, такий підхід вимагає декількох автентифікацій для сучасних додатків, які отримують доступ до ресурсів у різних системах.

Слабкі місця автентифікації на основі паролів можуть бути вирішені до певної міри більш складними іменами користувачів і правилами паролів, такими як мінімальна довжина та умови для складності, наприклад, включаючи великі і символи.

Інші способи автентифікації включають:

- Двофакторна автентифікація - додає додатковий рівень захисту процесу автентифікації. 2FA вимагає, щоб користувач надав другий фактор автентифікації додатоково до пароля.
- Багатофакторна автентифікація вимагає від користувачів автентифікації з більш ніж одним фактором автентифікації, включаючи біометричний фактор.
- Одноразовий пароль - це автоматично створений числовий або буквено-цифровий рядок символів, який використовується для

автентифікації користувача. Цей пароль дійсний лише для одного сеансу входу або транзакції.

● Біометрія - процес перевірки вашої ідентичності за допомогою ваших вимірювань або інших унікальних характеристик вашого тіла [13].

● Мобільна автентифікація - це процес перевірки користувача через їхні пристрої або перевірку самих пристроїв. Процес автентифікації мобільного зв'язку включає багатофакторну автентифікацію, яка може включати одноразові паролі, біометричну автентифікацію або перевірку QR-коду.

● Автентифікація API - стандартні методи керування автентифікацією API: HTTP basic autentification; Ключі API та OAuth.

● Відкрита авторизація (OAuth) - це відкритий стандарт для автентифікації та авторизації на основі маркерів в Інтернеті [5].

Програми-боти також повинні авторизувати свої автоматизовані дії в мережі. Інтернет-служби резервного копіювання, системи виправлення та оновлення та системи віддаленого моніторингу, які використовуються в телемедицині та технологіях інтелектуальних мереж, всі повинні надійно перевіряти автентичність, щоб переконатися, що це авторизована система, задіяна в будь-якій взаємодії, а не хакер.

Автоматична автентифікація може бути виконана з авторизаційними даними, подібними до ідентифікатора користувача та пароля, що надаються лише відповідною програмою. Вони також можуть використовувати цифрові сертифікати, видані та перевірені сертифікаційним органом, як частину інфраструктури відкритого ключа, щоб довести ідентифікацію під час обміну інформацією через Інтернет, як тип цифрового пароля.

Важливо розуміти, що кожна точка доступу є потенційною точкою вторгнення. Кожному мережевому пристрою потрібна сильна автентифікація.

1.6 Аналіз сучасних методів автентифікації

На сьогодні існує декілька способів автентифікації користувачів, у кожного з яких свої переваги і недоліки. Всі методи можна розділити на 4 великі групи, які наведені у таблиці 1.1.

Таблиця 1.1 Методи автентифікації користувачів

№	Методи автентифікації	Характеристика методу
1	Методи, засновані на знанні секретної інформації	Класичним прикладом таких методів є парольний захист, коли в якості засобу автентифікації користувачу пропонується ввести пароль – деяку послідовність символів. Такі методи автентифікації є найпоширенішими
2	Методи, засновані на використанні унікального предмета	В якості такого предмета можуть бути використані: смарт-карта, токен, електронний ключ тощо.
3	Методи, засновані на використанні біометричних характеристик людини	На практиці частіше використовуються одна або деякі з наступних біометричних характеристик: відбитки пальців (найбільш розповсюджено); малюнок сітківки або райдужної оболонки ока; термографія долоні; геометрія і термограма обличчя; почерк (підпис); голос
4	Методи, засновані на інформації, асоційованій з	Прикладом такої інформації можуть бути координати користувача, визначені за допомогою GPS. Цей підхід навряд чи

	користувачем	може бути використаний як єдиний механізм автентифікації, проте цілком допустимо його використання як додаткового елементу захисту

Парольні системи захисту. Головна перевага парольної ідентифікації - простота і звичність. Паролі давно вбудовані в операційні системи та інші сервіси. При правильному використанні паролі можуть забезпечити прийнятний для багатьох організацій рівень безпеки. Проте за сукупністю характеристик їх слід визнати найслабкішим засобом перевірки достовірності. Саме слабкий рівень парольного захисту є однією з основних причин вразливості комп'ютерних систем до спроб несанкціонованого доступу (НСД).

Хешування (використання незворотної хешфункції до будь-якої інформації перетворює її на унікальний код) не забезпечує захист від підбору паролів по словнику у разі отримання бази даних зловмисником. При виборі алгоритму хешування, який буде використаний для розрахунку згорток паролів, необхідно гарантувати неспівпадання значень згорток, отриманих на основі різних паролів користувачів. Крім того, слід передбачити механізм, що забезпечує унікальність згорток у випадку, якщо два користувачі вибирають однакові паролі. При шифруванні паролів особливе значення має спосіб генерації і зберігання ключа шифрування бази даних облікових записів. Перерахуємо деякі можливі варіанти: ключ генерується програмно і зберігається в системі, забезпечуючи можливість її автоматичного перезавантаження; ключ генерується програмно і зберігається на зовнішньому носієві, з якого прочитується при кожному запуску; ключ генерується на основі вибраного адміністратором пароля, який вводиться в систему при кожному запуску. Найбезпечніше зберігання паролів забезпечується при їх хешуванні і подальшому шифруванні отриманих згорток, тобто при їх

комбінації. Враховуючи, що користувачі нерідко вибирають недостатньо стійкі паролі, можна зробити висновок, що отримання бази даних облікових записів або перехоплення переданого по мережі значення згортки пароля представляють серйозну загрозу безпеці парольної системи. У захищеній системі передачу можна застосовувати тільки у поєднанні із засобами захисту мережевого трафіку.

Ідентифікація з використанням унікального предмета. Кожен апаратний (електронний) ідентифікатор є фізичним пристроєм (eToken), який зазвичай невеликих розмірів (його можна носити із собою), зручний та недорогий.

Основне призначення:

- двофакторна автентифікація користувачів при доступі до захищених ресурсів (комп'ютерів, мереж, додатків);
- безпечне зберігання закритих ключів цифрових сертифікатів, криптографічних ключів, профілів користувачів, налаштувань додатків тощо в незалежній пам'яті ключа;
- апаратне виконання криптографічних операцій в довіреному середовищі (генерація ключів шифрування, симетричне і асиметричне шифрування, розрахунок хеш-функції, формування електронного цифрового підпису - ЕЦП).

Можливі переваги застосування:

- сувора автентифікація користувачів при доступі до серверів, баз даних, розділів веб-сайтів;
- безпечне зберігання секретної інформації: паролів, ключів шифрування, закритих ключів цифрових сертифікатів;
- захист електронної пошти (цифровий підпис і шифрування, доступ);
- системи електронної торгівлі;
- захист комп'ютерів;
- захист мереж та каналів передачі даних за рахунок побудови - віртуальні приватні мережі.

Переваги eToken:

- автентифікація користувачів за рахунок використання криптографічних методів;
- безпечне зберігання ключів шифрування і ЕЦП;
- мобільність користувача і можливість безпечної роботи з конфіденційними даними в недовіреному середовищі (наприклад, на чужому комп'ютері);
- безпечне використання - скористатися ключем eToken може тільки його власник;
- реалізація як західних, так і вітчизняних стандартів на шифрування;
- зручність роботи - ключ виконаний у вигляді брелока зі світловою індикацією режимів роботи і безпосередньо підключається до USB-портів;
- використання одного ключа для вирішення безлічі завдань - входу в комп'ютер, входу в мережу, захисту каналу, шифрування інформації, ЕЦП.

Біометрична ідентифікація - це спосіб ідентифікації особи за окремими специфічними біометричними ознаками. Сучасний рівень розвитку комп'ютерних технологій дав змогу використовувати подібні ознаки як основу для ідентифікації людини і ухвалення рішення про доступ до ресурсів [14]. При всьому теоретичному різноманітті можливих біометричних методів тих, що застосовуються на практиці серед них небагато. Основних методів три:

- за відбитком пальця;
- за зображенням особи (двомірному або тривимірному);
- за райдужною оболонкою та за сітківкою ока.

1.7 One tap Sign-up/ Auto sign-in

Новий метод входу дозволяє користувачам реєструватися, не турбуючи їх екраном реєстрації. Користувачі отримують на вашому сайті безпечний обліковий запис на основі маркерів, без паролів, захищений обліковим записом Google.

Програма тестування бета-версії для цього API зараз закрита.

Управління федеративними ідентифікаціями (FIM) - це угода, яка може бути зроблена між декількома підприємствами, щоб дозволити абонентам використовувати ті ж ідентифікаційні дані, щоб отримати доступ до мереж усіх підприємств групи. Використання такої системи іноді називається федерацією ідентифікації.

Федерація ідентифікації пов'язує ідентифікацію користувача з кількома доменами безпеки, кожен з яких підтримує свою власну систему управління ідентифікацією. Коли два домени об'єднані, користувач може пройти автентифікацію в одному домені, а потім отримати доступ до ресурсів в іншому домені без необхідності виконання окремого процесу входу.

Федерація ідентифікації пропонує економічні переваги, а також зручність для підприємств та їхніх абонентів. Наприклад, кілька корпорацій можуть спільно використовувати одну заявку, що призводить до економії коштів і консолідації ресурсів.

Це новий шлях для користувачів увійти або зареєструватись на веб-сайті з одним натисканням комп’ютерної миші, використовуючи аккаунт, який є на третьому веб-ресурсі, що називається identity provider.

Identity federation побудована на стандартах коду, таких як OpenID Connect та OWS. З цією технологією, користувачам не потрібно створювати додатковий пароль.

Redfin - компанія в США, яка займається нерухомістю , спостерігала результат збільшення кількості реєстрації нових користувачіва на 80% після впровадження One tap Sign-up/ Auto sign-in. Крім того, більше 40% нових користувачів повернулися на веб-сайт більше п'яти разів після реєстрації.

Компанія Trivago є одним з світових лідерів у пошуку та замовленню готелів, що працює у 55 країнах. Вона набрала на 50% більше нових користувачів після реалізації цієї бібліотеки.

Cifroclub, Letras, і Palco - популярні музичні сайти в Бразилії для пісень та текстів, отримали в 43 рази більше зареєстрованих користувачів після інтеграції One tap Sign-up/ Auto sign-in.

1.8 Автентифікація за сертифікатами

Сертифікат являє собою набір атрибутів, що ідентифікують власника, підписаний certificate authority (CA). CA виступає в ролі посередника, який гарантує справжність сертифікатів (за аналогією з ФМС, що випускає паспорта). Також сертифікат криптографічно пов'язаний з закритим ключем, який зберігається у власника сертифіката і дозволяє однозначно підтвердити факт володіння сертифікатом.

На стороні клієнта сертифікат разом з закритим ключем можуть зберігатися в операційній системі, в браузері, в файлі, на окремому фізичному пристрої (smart card, USB token). Зазвичай закритий ключ додатково захищений паролем або PIN-кодом.

У веб-додатках традиційно використовують сертифікати стандарту X.509. Автентифікація за допомогою X.509-сертифіката відбувається в момент з'єднання з сервером і є частиною протоколу SSL / TLS. Цей механізм також добре підтримується браузерами, які дозволяють користувачеві вибрати і застосувати сертифікат, якщо веб-сайт допускає такий спосіб автентифікації.

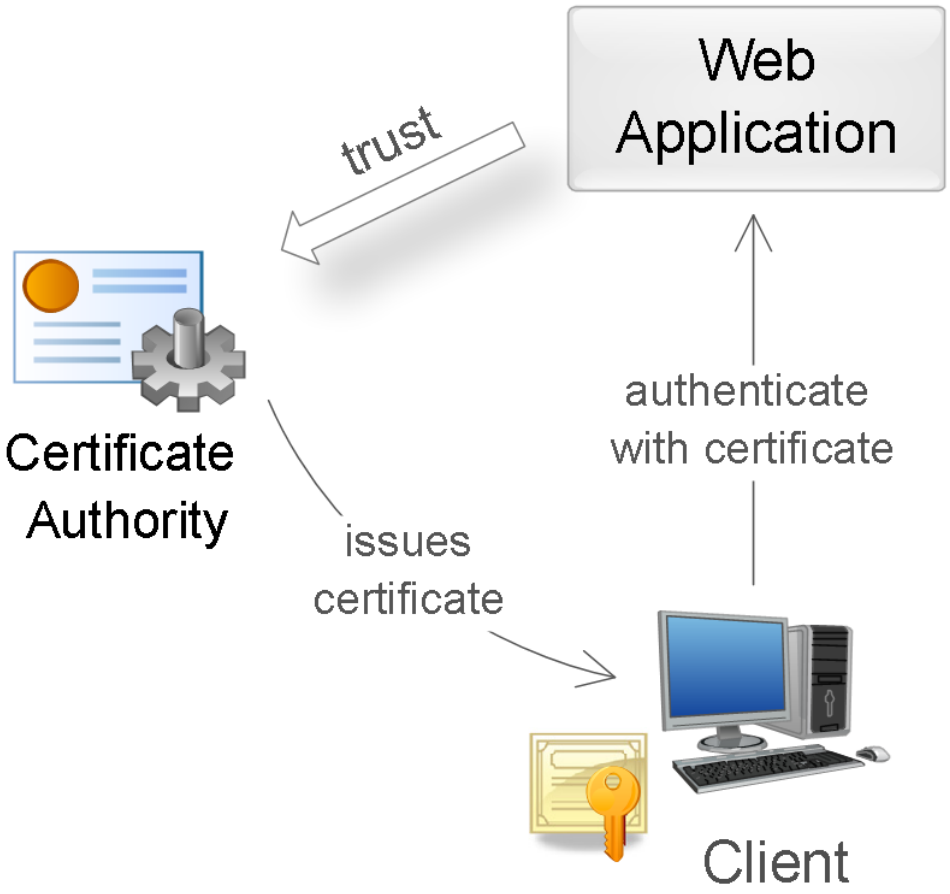

Рисунок 1.2 Використання сертифікату для автентифікації

Під час автентифікації сервер виконує перевірку сертифіката на підставі наступних правил:

1. Сертифікат повинен бути підписаний.
2. Сертифікат повинен бути дійсним на поточну дату (перевірка терміну дії).
3. Сертифікат не повинен бути відкликаний відповідним CA (перевірка списків виключення).

Використання сертифікатів для автентифікації - куди більш надійний спосіб, ніж автентифікація за допомогою паролів. Це досягається створенням в процесі автентифікації цифрового підпису, наявність якої доводить факт застосування закритого ключа в конкретній ситуації (non-repudiation). Однак труднощі з поширенням і підтримкою сертифікатів робить такий спосіб

автентифікації малодоступним в широких колах.На основі проаналізованих даних можна скласти наступну порівняльну таблицю:

Таблиця 1.2 Порівняльна характеристика методів автентифікації

Метод	Популярність	Ціна інфраструктури	Простота використання	Рівень безпеки	Переваги	Недоліки
Простий статичний пароль	******	*	****	*	Дуже низька вартість, портативність, знайомі користувачам.	Дуже низький рівень безпеки, зручність користування зменшується з кількома паролями.
Сильний статичний пароль	****	**	*	**	Краща безпека, ніж простий пароль	Зручність користування, вразливість до відтворення.
За сертиіфкатом	***	*****	****	****	Офлайн-фізична автентифікація.	Витрати на інфраструктуру, вартість маркер.
За одноразовим паролем	***	***	****	****	Хороша безпека, хороша зручність використання.	Форм-фактор.
Біометричний	*	*****	*****	*****	Хороша загальна	Неможливість зміни

					безпека.	параметрів, переважно висока вартість.

1.9 Висновки до розділу

У цьому розділі було розглянуто процес автентифікації, а також основні методи та нормативно-правову базу. Звідки можна зробити висновок, що автентифікація є важливим процесом будь-якої системи, оскільки вона дозволяє організаціям підтримувати безпеку своїх мереж, дозволяючи лише авторизованим користувачам отримувати доступ до захищених ресурсів, які можуть включати комп'ютерні системи, мережі, бази даних, веб-сайти та інші мережеві програми або служби.

Авторизація включає в себе процес, за допомогою якого адміністратор надає права автентифікованим користувачам, а також процес перевірки дозволів користувача облікового запису, щоб переконатися, що користувачеві було надано доступ до цих ресурсів. Привілеї та переваги, надані авторизованому обліковому запису, залежать від дозволів користувача, які зберігаються локально або на сервері автентифікації.

Діяльність із захисту інформації регламентується законами України, до яких належать такі закони, як: «Про інформацію», «Про захист інформації в інформаційно-телекомунікаційних системах», «Про захист персональних даних», «Про електронні довірчі послуги», «Про основні засади забезпечення кібербезпеки України»; а також нормативно–правовими актами із ТЗІ: «Критерії оцінки захищеності інформації в комп’ютерних системах від НСД», «Загальні положення щодо захисту інформації в комп'ютерних системах від НСД» та «Вимоги до засобів електронної ідентифікації, рівнів довіри до засобів електронної ідентифікації для їх використання у сфері електронного урядування».

Автентифікація є невід’ємною частиною функціонування будь-якого веб-додатку і вразливості пов’язані з нею є актуальними та критичними для більшості сучасних систем.

РОЗДІЛ 2
ВРАЗЛИВОСТІ ВЕБ-ДОДАТКІВ ПОВ'ЯЗАНІ З АВТЕНТИФІКАЦІЄЮ ТА КОНТРОЛЕМ ДОСТУПУ. ОСОБЛИВОСТІ ЗАХИСТУ.

2.1 Вразливості і недоліки автентифікації по паролю

Автентифікація по паролю вважається не дуже надійним способом, так як пароль часто можна підібрати, а користувачі схильні використовувати прості і однакові паролі в різних системах, або записувати їх на клаптиках паперу. Якщо зловмисник зміг з'ясувати пароль, то користувач часто про це не дізнається. Крім того, розробники додатків можуть допустити ряд концептуальних помилок, що спрощують злом облікових записів.

Нижче представлений список найбільш часто зустрічаються вразливостей в разі використання автентифікації по паролю:

- Веб-додаток дозволяє користувачам створювати прості паролі.
- Веб-додаток не захищений від можливості перебору паролів (brute-force attacks).
- Веб-додаток саме генерує і поширює паролі користувачам, однак не вимагає зміни пароля після першого входу (тобто поточний пароль десь записаний).
- Веб-додаток допускає передачу паролів по незахищеному HTTP-з'єднання, або в рядку URL.
- Веб-додаток не використовує безпечні хеш-функції для зберігання паролів користувачів.
- Веб-додаток не надає користувачам можливість зміни пароля або не нотифікує користувачів про зміну їх паролів.

- Веб-додаток використовує вразливу функцію відновлення пароля, яку можна використовувати для отримання несанкціонованого доступу до інших облікових записів.
- Веб-додаток не вимагає повторної автентифікації користувача для важливих дій: зміна пароля, зміни адреси доставки товарів і т. П.
- Веб-додаток створює session tokens таким чином, що вони можуть бути підібрані або передбачені для інших користувачів.
- Веб-додаток допускає передачу session tokens по незахищеному HTTP-з'єднання, або в рядку URL.
- Веб-додаток вразливий для session fixation-атак.
- Веб-додаток не встановлює прапори HttpOnly і Secure для browser cookies, що містять session tokens.
- Веб-додаток не знищує сесії користувача після короткого періоду неактивності або не надає функцію виходу з сесії [10].

2.1 Підбір (Brute Force)

Підбір - автоматизований процес проб і помилок, що використовується для того, щоб вгадати ім'я користувача, пароль, номер кредитної картки, ключ шифрування і т.д.

Багато систем дозволяють використовувати слабкі паролі або ключі шифрування, і користувачі часто вибираю легко вгадувані або такі, що містяться у словниках парольні фрази.

Використовуючи цю ситуацію, зловмисник може скористатися словником і спробувати використати тисячі або навіть мільйони комбінацій символів в якості пароля. Якщо випробуваний пароль дозволяє отримати

доступ до системи, атака вважається успішною і атакуючий може використовувати обліковий запис.

Подібна техніка проб і помилок може бути використана для підбору ключів шифрування. У разі використання ключів недостатньою довжини, зловмисник може отримати необхідний ключ, перебравши всі можливі комбінації.

Існує два види підбору: прямий і зворотний. При прямому підборі використовуються різні варіанти пароля для одного імені користувача. При зворотному перебираються різні імена користувачів, а пароль залишається незмінним. У системах з мільйонами облікових записів ймовірність використання різними користувачами одного пароля досить висока. Не дивлячись на популярність і високу ефективність, підбір може займати кілька годин, днів або років.

Приклад:

Ім’я користувача = Jon

Пароль = smith, michael-jordan, [pet names], [birthdays], [car names], ...

Імена користувачів = Jon, Dan, Ed, Sara, Barbara, ...

Пароль = 12345678

2.2 Недостатня аутентифікація (Insufficient Authentication)

Ця вразливість виникає, коли Web-сервер дозволяє атакуючому отримувати доступ до важливої інформації або функцій сервера без належної автентифікації. Інтерфейси адміністрування через Web - яскравий приклад критичних систем.

Залежно від специфіки програми, подібні компоненти не повинні бути доступні без належної автентифікації. Щоб не використовувати аутентифікацію деякі ресурси "ховаються" за певною адресою, на яку немає посилань з основних сторінках сервера або інших загальнодоступних ресурсах. Однак, подібний підхід не більш ніж "безпека через приховування".

Важливо розуміти, що, не дивлячись на те, що зловмисник не знає адреси сторінки, вона все одно доступна через Web.

Необхідний URL може бути знайдений перебором типових файлів і директорій (таких як / admin /), з використанням повідомлень про помилки, журналів перехресних посилань або шляхом простого читання документації. Подібні ресурси повинні бути захищені адекватно важливості їх вмісту і функціональних можливостей [19].

Багато Web-додатків за замовчуванням використовують для адміністративного доступу посилання в кореневій директорії сервера (/ admin /) [4]. Зазвичай посилання на цю сторінку не фігурує у вмісті сервера, проте сторінка доступна за допомогою стандартного браузера. Оскільки користувач або розробник припускає, що ніхто не скористається цією сторінкою, так як посилання на неї відсутнє, найчастіше реалізацією автентифікації нехтують. І для отримання контролю над сервером зловмисникові достатньо зайти на цю сторінку.

2.3 Небезпечне відновлення паролів (Weak Password Recovery Validation)

Ця вразливість виникає, коли Web-сервер дозволяє атакуючому несанкціоновано отримувати, модифікувати або відновлювати паролі інших користувачів.

Часто аутентифікація на Web-сервер вимагає від користувача запам'ятовування пароля або парольної фрази. Тільки користувач повинен знати пароль, причому пам'ятати його виразно. З часом пароль забувається. Ситуація ускладнюється, оскільки в середньому користувач відвідує близько 20 сайтів, що вимагають введення пароля.

Таким чином, функція відновлення пароля є важливою складовою що надається Web-серверами сервісу.

Прикладом реалізації подібної функції є використання "секретного питання", відповідь на яке вказується в процесі реєстрації. Питання або

вибирається зі списку або вводиться самим користувачем. Ще один механізм дозволяє користувачу вказати "підказку", яка допоможе йому згадати пароль. Інші способи вимагають від користувача вказати частину персональних даних, таких як номер соц. страхування, ІПН, домашню адресу, поштовий індекс і т.д., які потім будуть використовуватися для встановлення особи. Після того як користувач доведе свою ідентичність, система відобразить новий пароль або перешле його поштою.

Вразливості пов'язані з недостатньою перевіркою при відновленні паролю виникають, коли атакуючий отримує можливість використовувати даний механізм. Це трапляється, коли інформацію, використовувану для перевірки користувача, легко вгадати або сам процес підтвердження можна обійти. Система відновлення пароля може бути скомпрометована шляхом використання підбору, вразливостей системи або вгадування відповіді на секретне питання.

Багато серверів вимагають від користувача вказати його email в комбінації з домашньою адресою і номером телефону. Ця інформація може бути легко отримана з мережевих довідників. В результаті, дані, які використовуються для перевірки, не є великим секретом. Крім того, ця інформація може бути отримана зловмисником з використанням інших методів, таких як міжсайтового виконання сценаріїв або фішингу [5].

Парольні підказки. Сервер, що використовує підказки для полегшення запам'ятовування паролів, може бути атакований, оскільки підказки допомагають у реалізації підбору паролів. Користувач може використовувати стійкий пароль, наприклад, "221277King" з відповідною підказкою: "д-р + улюблений письменник". Атакуючий може припустити, що для користувача пароль складається з дати народження та імені улюбленого автора користувача. Це допомагає сформувати відносно короткий словник для атаки шляхом перебору.

Таємне питання та відповідь. Припустимо, відповідь користувача "Бобруйськ", а секретне питання "Місце народження". Зловмисник може

обмежити словник для підбору секретної відповіді назвами міст. Більш того, якщо атакуючий має в своєму розпорядженні деяку інформацію про користувача, дізнатися його місце народження не складно.

2.4 Відсутність таймауту сесії (Insufficient Session Expiration)

У випадку якщо для ідентифікатору сесії або облікових даних не передбачений таймаут або має значення дуже велике, зловмисник може скористатися старими даними для авторизації. Це підвищує вразливість сервера для атак, пов'язаних з крадіжкою ідентифікаційних даних. Оскільки протокол HTTP не передбачає контроль сесії, Web-сервери зазвичай використовують ідентифікатори сесії для визначення запитів користувача. Таким чином, конфіденційність кожного ідентифікатора повинна бути забезпечена, щоб запобігти багаторазовий доступ користувачів з одним профілем. Викрадений ідентифікатор може використовуватися для доступу до даних користувача або здійснення шахрайських транзакцій. Відсутність таймауту сесії збільшує ймовірність успіху різних атак. Приміром, зловмисник може отримати ідентифікатор сесії, використовуючи мережевий аналізатор або вразливість типу міжсайтового виконання сценаріїв. Хоча таймаут не допоможе у випадку, якщо ідентифікатор буде використаний негайно, обмеження часу допоможе у випадку більш пізніх спроб використання ідентифікатора.

В іншій ситуації, якщо користувач отримує доступ до сервера з публічного комп'ютера (бібліотека, Internet-кафе і т.д.) відсутність таймауту сесії може дозволити зловмисникові скористатися історією браузера для перегляду сторінок користувача.

Велике значення таймауту збільшує шанси підбору чинного ідентифікатора. Крім того, збільшення цього параметра веде до збільшення одночасно відкритих сесій, що ще більше підвищує ймовірність успішного підбору.

При використанні публічного комп'ютера, коли кілька користувачів мають необмежений фізичний доступ до машини, відсутність таймауту сесії дозволяє зловмисникові переглядати сторінки, відвідані іншим користувачем. Якщо функція виходу з системи просто перенаправляє на основну сторінку Web-сервера, а не завершує сесію, сторінки, відкриті користувачем, можуть бути переглянуті зловмисником. Оскільки ідентифікатор сесії не був відзначений як недійсний, атакуючий отримає доступ до сторінок сервера без повторної автентифікації [6, 7].

2.5 Фіксація сесії (Session Fixation)

Використовуючи даний клас атак, зловмисник присвоює ідентифікатору сесії користувача задане значення. Залежно від функціональних можливостей сервера, існує декілька способів "зафіксувати" значення ідентифікатора сесії. Для цього можуть використовуватися атаки типу міжсайтового виконання сценаріїв або підготовка сайту з допомогою попереднього HTTP запиту. Після фіксації значення ідентифікатора сесії атакуючий очікує моменту, коли користувач увійде в систему. Після входу користувача, зловмисник використовує ідентифікатор сесії для отримання доступу до системи від імені користувача.

Можна виділити два типи систем управління сесіями на основі ідентифікаторів. Перший з них, "дозволяючий", дає змогу браузеру вказувати будь-який ідентифікатор. Системи другого "суворого" типу обробляють тільки ідентифікатори, згенеровані сервером. Якщо використовуються "дозволяючі" системи, зловмисник може вибрати будь-який ідентифікатор сесії. У випадку із "суворими" серверами зловмисникові доводиться підтримувати "сесію-заглушку" і періодично з'єднуватися з сервером для уникнення закриття сесії за таймаут.

Без наявності активного захисту від фіксації сесії, ця атака може бути використана проти будь-якого сервера, який аутентифікує користувачів за

допомогою ідентифікатора сесії. Більшість Web-серверів зберігає ID в cookie, але це значення так само може бути присутнім в URL або прихованому полі форми. На жаль, системи, що використовують cookie, є найбільш вразливими. Більшість відомих на даний момент варіантів фіксації сесії спрямовані саме на значення cookie. На відміну від крадіжки ідентифікатора, фіксація сесії надає зловмисникові набагато більший простір для творчості. Це пов'язано з тим, що активна фаза атаки відбувається до входу користувача в систему.

Атаки, спрямовані на фіксацію сесії зазвичай проходять у три етапи:

1. Встановлення сесії. Зловмисник встановлює сесію-заглушку на атакуючому сервер і отримує від сервера ідентифікатор або вибирає довільний ідентифікатор. У деяких випадках сесія-заглушка повинна підтримуватися в активному стані шляхом періодичних звернень до сервера.
2. Фіксація сесії. Зловмисник передає значення ідентифікатора сесії-заглушки браузеру користувача та фіксує його ідентифікатор сесії. Це можна зробити, наприклад, встановивши значення cookie в браузері за допомогою XSS.
3. Підключення до сесії. Атакуючий очікує автентифікації користувача на сервері. Після того, як користувач зайшов на сайт, зловмисник підключається до сервера, використовуючи зафіксований ідентифікатор, і отримує доступ до сесії користувача.

Для фіксації ID сесії можуть бути використані різні техніки, такі як:

- Установка значення cookie за допомогою мов сценаріїв на стороні клієнта. Якщо вразливість типу міжсайтового виконання сценаріїв присутня на будь-якому сервері в домені, зловмисник отримує можливість встановити значення cookie на стороні клієнта. (Приклад коду:http://example/ <script> document.cookie ="sessionid = 1234;% 20domain =.example.dom"; </ script>. idc)

- Установка значення cookie за допомогою тега META. Це техніка схожа на попередню, але може бути використана, коли вжито заходів проти впровадження тегів сценаріїв. (Приклад коду: http://example/<meta%20http-equiv=SetCookie%20content="sessionid=1234;%20domain=.example.dom">. idc).
- Установка cookie з використанням заголовка відповіді HTTP. Зловмисник використовує атакуючий сервер або будь-який сервер у домені для того, щоб встановити cookie з ідентифікатором сесії. Це може бути реалізовано різними методами, наприклад:
    - Злом сервера в домені (наприклад, слабо адмініструються сервери WAP).
    - Підміна значень в кеші DNS-сервера користувача з метою додати атакуючий сервер в домен.
    - Установка помилкового WEB-сервера в домені (наприклад, на робочій станції в середовищі Active Directory, де всі машини в DNS належать одному домену).
    - Використання атаки типу розщеплення HTTP відповіді (response splitting).

Зауваження: Фіксація сесії на тривалий проміжок часу може бути здійснено з використанням постійних cookie (наприклад, з терміном дії 10 років), які зберігаються навіть після перезавантаження комп'ютера.

2.6 Особливості захисту інформації у веб-додатках

На сьогоднішній день, майже всі брандмауери веб-ресурсів покликані захистити від основних типів загроз. А саме:

- SQL ін'єкція; – міжсайтовий скриптинг (XSS);
- міжсайтова підробки запитів (CSRF);

- розподілена відмова в обслуговуванні (DDoS- атаки,);
- відсутність таймаута сесії;
- зворотний шлях в директоріях. Загрозам піддаються і WEB-сервери.

Класифікація атак на WEB-сервер має ієрархічну структуру та розділяється на шість основних класів. А саме:

- атаки на засоби автентифікації;
- атаки на засоби авторизації;
- атаки на клієнтів;
- атаки направлені на виконання коду;
- атаки направлені на розголошення інформації;
- логічні атаки.

Захист Web-ресурсів від несанкціонованого доступу призначений для криптографічного захисту та розмежування доступу до інформації, оброблюваної в ІС, побудованих на базі Web-технологій.

При цьому забезпечується:

- взаємна автентифікація (підтвердження справжності) клієнта та сервера за протоколом, побудованим із використанням не симетричних криптографічних алгоритмів;
- захист конфіденційності та цілісності інформації, що передається між клієнтом та сервером, з використанням алгоритмів симетричного шифрування/ дешифрування інформації та вироблення/перевірки кодів автентифікації повідомлень;
- розмежування доступу користувачів до інформаційних ресурсів, представлених у вигляді статичних або динамічних Web-сторінок, що зберігаються та оброблюються на відповідних Web-серверах та потребують захисту (захищених Web-ресурсів).

Захист Web-ресурсів повинен реалізувати такі основні функції:

- ідентифікацію та автентифікацію користувачів на основі атрибутів, отриманих від серверу автентифікації, що дозволяє

однозначно встановити певного користувача та у подальшому коректно оброблювати його запити на доступ до захищеної інформації або до засобів адміністрування;

- виділення, на підставі результатів виконаної автентифікації, користувачів-адміністраторів, яким надані певні повноваження;
- сувору взаємну автентифікацію клієнтського та серверного компонентів системи та їхніх користувачів з використанням відповідних протоколів, у яких використовується механізм вироблення/перевірки електронного цифрового підпису (ЕЦП) за алгоритмом з використанням відповідних атрибутів (особистих ключів (ОК) та сертифікатів відкритих ключів (ВК) відповідних користувачів);
- керування доступом користувачів до захищених Web-ресурсів (окремих Web-сторінок та їх сукупностей), представлених відповідними URL-адресами, на основі атрибутів доступу, призначених спеціально вповноваженими адміністраторами;
- шифрування/дешифрування інформації, що передається між Web-браузером (або Web-застосуванням), який функціонує на робочій станції (РС) користувача, та захищеним Web-сервером (Proxy-сервером), що забезпечує захист конфіденційності інформації на всьому шляху її передачі по каналах мережі Internet;
- контроль цілісності інформації, що передається між Web-браузером (Web-додатком), яке функціонує на РС користувача, та захищеним Web-сервером (Proxy- сервером), що забезпечує захист від її несанкціонованої модифікації на всьому шляху її передачі по мережі Internet;
- контроль цілісності та самотестування програмних засобів системи при старті та в процесі функціонування, що дозволяє забезпечити стійке функціонування засобів захисту та не

допустити обробки повідомлень у випадку порушення працездатності.

Правила розмежування доступу (access mediation rules) — частина політики безпеки, що регламентує правила доступу користувачів і процесів до пасивних об’єктів.

При розгляді взаємодії двох об'єктів комп’ютерної системи, що виступають як приймальники або джерела інформації, слід виділити пасивний об'єкт, над яким виконується операція, і активний об'єкт, який виконує або ініціює цю операцію.

Коли користувачі або процеси намагаються одержати доступ до пасивних об'єктів, механізми, що реалізують керування доступом, на підставі політики безпеки і перевірки атрибутів доступу можуть «прийняти рішення» про легальність запиту. Використовуючи набір атрибутів доступу відповідно до прийнятої політики безпеки, можна реалізувати довірче керування доступом, адміністративне, контроль за цілісністю та інші види керування доступом.

Для відображення функціональності комп’ютерної системи використовується концепція матриці доступу. Матриця доступу являє собою таблицю, уздовж кожного виміру якої відкладені ідентифікатори об'єктів комп’ютерної системи, а елементами матриці виступають дозволені або заборонені режими доступу. Матриця доступу може бути двомірною (наприклад, користувачі/пасивні об'єкти або процеси/пасивні об'єкти) або тримірною (користувачі/процеси/пасивні об'єкти) [4].

2.7 Висновки до розділу

У даному розділі було проаналізовано вразливості до веб-додатків пов’язані з процесом автентифікації, а також розглянуто основні методи захисту веб-додатків.

Значення WEB-додатків в наш час посідає значне місце, адже є одним з найефективніших інструментів, що дозволяє швидко та незалежно від місцезнаходження здійснювати операції, пов’язані з різними аспектами комерційної діяльності. Внаслідок чого існує неперервний попит на здійснення атак з метою отримання конфіденційних даних, що містять комерційну таємницю чи іншу важливу інформацію.

Проаналізувавши основні типи атак на WEB-додатки, можна зробити висновок, що для реалізації захисту WEB-додатку потрібно відштовхуватись від кожного додатку індивідуально, щоб вірно спрогнозувати основний вектор атаки, що буде здійснюватися та реалізовувати захист спираючись на отримані дані, застосовуючи комплекс різних методів захисту для досягнення більшої стійкості та безпеки додатку.

РОЗДІЛ 3
АВТЕНТИФІКАЦІЯ РІЗНИХ ТИПІВ КОРИСТУВАЧІВ ДО ВЕБ-ДОДАТКУ. АЛГОРИТМ РОЗМЕЖУВАННЯ ДОСТУПУ. ЗБЕРІГАННЯ ДАНИХ КОРИСТУВАЧІВ.

3.1 Автентифікація по паролю

В якості першого кроку процесу автентифікації користувачів у веб-додатку використано метод автентифікації по паролю. Цей метод грунтується на тому, що користувач повинен надати username і password для успішної ідентифікації і автентифікації в системі. Пара username / password задається користувачем при його реєстрації в системі, при цьому в якості username може виступати адреса електронної пошти користувача.

Протокол що використовується для автентифікації по паролю, описаний в стандартах HTTP 1.0 / 1.1, існує дуже давно і до цих пір активно застосовується в корпоративному середовищі. Працює за наступним принципом: в веб-додаток включається HTML-форма, в яку користувач повинен ввести свої username / password і відправити їх на сервер через HTTP POST для автентифікації. У разі успіху веб-додаток створює session token, який зазвичай поміщається в browser cookies. При наступних веб-запитах session token автоматично передається на сервер і дозволяє додатку отримати інформацію про поточного користувача для авторизації запиту.

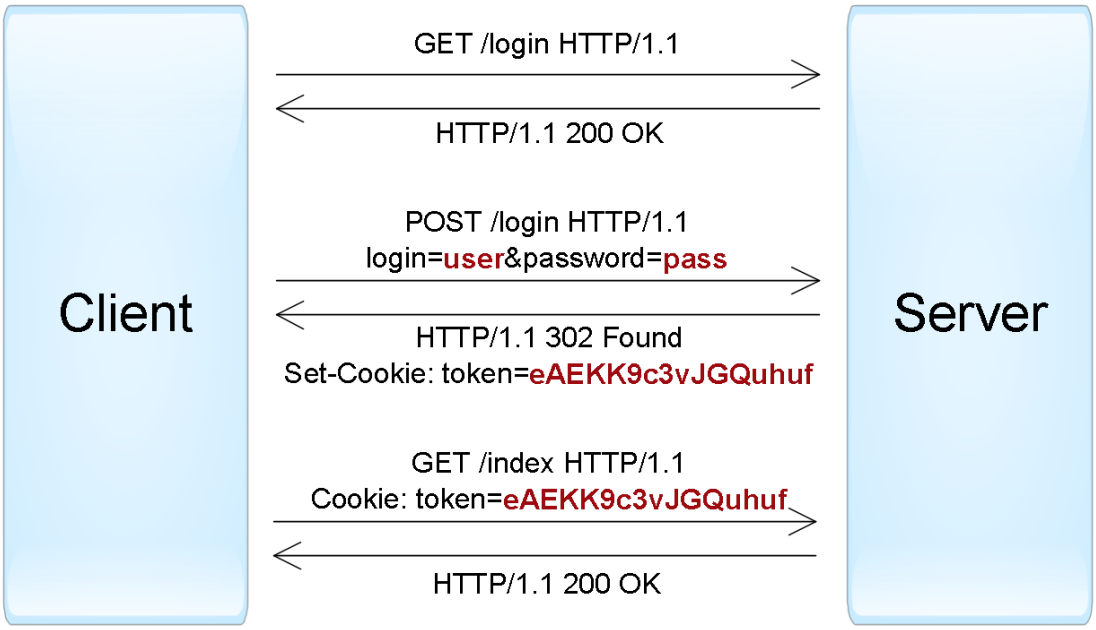

Рисунок 3.1 Приклад forms authentication

Додаток може створити session token двома способами:

1. Як ідентифікатор автентифікованої сесії користувача, яка зберігається в пам'яті сервера або в базі даних. Сесія повинна містити всю необхідну інформацію про користувача для можливості авторизації його запитів.

2. Як зашифрований і / або підписаний об'єкт, що містить дані про користувача, а також вказано термін. Цей підхід дозволяє реалізувати stateless-архітектуру сервера, однак вимагає механізму поновлення сесійного токена після закінчення терміну дії. Кілька стандартних форматів таких токенов розглядаються в секції «Автентифікація по токену».

Перехоплення session token часто дає аналогічний рівень доступу, що і знання username / password. Тому всі комунікації між клієнтом і сервером у разі forms authentication повинні проводитися тільки по захищеному з'єднанню HTTPS.

3.2 Додатковий крок автентифікації

Автентифікація за одноразовими паролями зазвичай застосовується додатково до автентифікації паролів для реалізації 2FA. У цій концепції користувачеві необхідно надати дані двох типів для входу в систему: щось, що він знає (наприклад, пароль), і щось, чим він володіє (наприклад, пристрій для генерації одноразових паролів). У розробленій системі тим що знає користувач є функція автентифікації яку він обирає у налаштуваннях профілю. Після підставлення згенерованого одноразового числа користувач вводить результат в якості відповіді на другий крок автентифікації.

Наявність двох факторів дозволяє в значній мірі збільшити рівень безпеки, що м. Б. затребуване для певних видів веб-додатків.

Інший популярний сценарій використання одноразових паролів - додаткова автентифікація користувача під час виконання важливих дій: переказ грошей, зміна налаштувань і т. д.

Існують різні джерела для створення одноразових паролів. Найбільш популярні:

1. Апаратні або програмні маркери, які можуть генерувати одноразові паролі на підставі секретного ключа, введеного в них, і поточного часу. Секретні ключі користувачів, які є фактором володіння, також зберігаються на сервері, що дозволяє виконати перевірку введених одноразових паролів. Приклад апаратної реалізацій токенов - RSA SecurID ; програмної - додаток Google Authenticator .

2. Випадково генеруються коди, що передаються користувачеві через SMS або інший канал зв'язку. У цій ситуації фактор володіння - телефон користувача (точніше - SIM-карта, прив'язана до певного номера).

3. Роздруківка або scratch card зі списком заздалегідь сформованих одноразових паролів. Для кожного нового входу в систему потрібно ввести новий одноразовий пароль з зазначеним номером.

У веб-додатках такий механізм автентифікації часто реалізується за допомогою розширення forms authentication: після первинної автентифікації по паролю, створюється сесія користувача, проте в контексті цієї сесії користувач не має доступу до додатка до тих пір, поки він не виконує додаткову аутентифікацію за одноразовим паролем.

3.3 Автентифікація “простих” користувачів

При автентифікації звичайних користувачів та полегшення процесу використання веб-додатку доцільно використовувати автентифікацію за токеном. Такий спосіб автентифікації найчастіше застосовується при побудові розподілених систем Single Sign-On (SSO), де один додаток (service provider або relying party) делегує функцію автентифікації користувачів іншому додатку (identity provider або authentication service). Типовий приклад цього способу - вхід в додаток через обліковий запис в соціальних мережах. Тут соціальні мережі є сервісами автентифікації, а додаток довіряє функцію автентифікації користувачів соціальних мереж.

Реалізація цього способу полягає в тому, що identity provider (IP) надає достовірні відомості про користувача в вигляді токена, A service provider (SP) додаток використовує цей токен для ідентифікації, автентифікаціїі авторизації користувача.

На загальному рівні, весь процес виглядає наступним чином:

1. Клієнт автентифікується в identity provider одним із способів, специфічним для нього (пароль, ключ доступу, сертифікат, Kerberos, ітд.).

2. Клієнт просить identity provider надати йому токен для конкретного SP-додатки. Identity provider генерує токен і відправляє його клієнту.

3. Клієнт автентифікуючий в SP-додатку за допомогою цього токена.

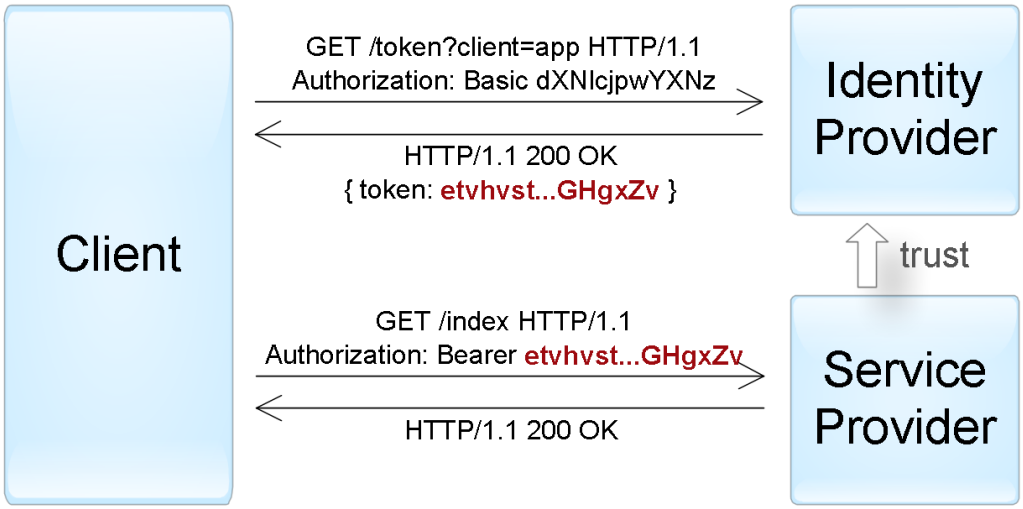

Рисунок 3.2 Автентифікація «активного» клієнта за допомогою токена, переданого за допомогою Bearer схеми.

Процес, описаний вище, відображає механізм автентифікації активного клієнта, тобто такого, який може виконувати запрограмовану послідовність дій (наприклад, iOS / Android програми). Браузер ж - пасивний клієнт в тому сенсі, що він тільки може відображати сторінки, запитані користувачем. В цьому випадку автентифікація досягається за допомогою автоматичного перенаправлення браузера між веб-додатками identity provider і service provider.

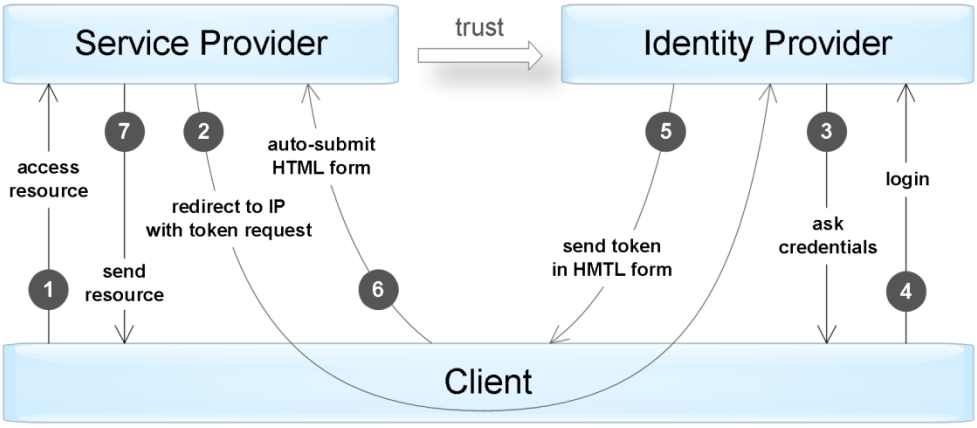

Рисунок 3.3 Автентифікація «пасивного» клієнта за допомогою перенаправлення запитів

Сам токен являє собою структуру даних, яка містить інформацію, хто згенерував токен, хто може бути одержувачем токена, термін дії, набір відомостей про самого користувача (claims). Крім того, токен додатково підписується для запобігання несанкціонованих змін і гарантій автентичності.

При автентифікації за допомогою токена SP-додаток повинен виконати наступні перевірки:

1. Токен був виданий identity provider додатком (перевірка поля issuer).
2. Токен призначається поточному SP-додатком (перевірка поля audience).
3. Термін дії токена ще не закінчився (перевірка поля expiration date).
4. Токен справжній і не був змінений (перевірка підпису).

У разі успішної перевірки SP-додаток виконує авторизацію запиту на підставі даних про користувача, що містяться в токені. Для реалізації автентифікації за токеном використано стандарт JSON Web Token.

JSON Web Token (JWT) - містить три блоки, між якими ставлять крапку: заголовок, набір полів (claims) і підпис. Перші два блоки представлені в JSON-форматі і додатково закодовані в формат base64. Набір полів містить довільні пари ім'я / значення, до того ж стандарт JWT визначає кілька зарезервованих імен (iss, aud, exp і інші). Підпис може генеруватися за допомогою і симетричних алгоритмів шифрування, і асиметричних. Крім того, існує окремий стандарт, відписувався формат зашифрованого JWT-токена [12].

Приклад підписаного JWT токена (після декодування 1 і 2 блоків):

{ «Alg»: «HS256», «typ»: «JWT»}.
{ «Iss»: « auth.myservice.com », «aud»: « myservice.com », «exp»:

«1435937883», «userName»: «John Smith», «userRole»: «Admin»}. S9Zs / 8 / uEGGTVVtLggFTizCsMtwOJnRhjaQ2BMUQhcY

3.4 Алгоритм авторизації у веб-додатку

Коли користувачі вперше відвідують WEB-сайт, вони анонімні. За замовчуванням анонімні користувачі не можуть звертатися до будь-якої сторінки WEB-додатку. Схема, що відображає етапи виконання алгоритму контролю доступу програмного забезпечення. Система захисту WEB-додатку працює за таким алгоритмом:

1. Запит WEB-додатку. Користувач відправляє запит на WEB-сервер. Оскільки ідентичність користувача в цей момент не відома, йому пропонується зареєструватися, використовуючи спеціальну WEB-сторінку.

2. Користувач надає інформацію, яка обробляється WEB-додатком для авторизації користувача у системі. WEB-сервер обробляє надані користувачем дані:

    - якщо інформація користувача підтверджується, йому надається доступ до WEB-сторінки;

    - якщо інформація користувача оцінюється як некоректною, йому пропонується повторити спробу реєстрації, або ж виконується переадресація на сторінку з повідомленням про закриття доступу.

Розподілення ролей в системі. Ролі ідентифікованого користувача порівнюються з списком дозволених користувачів і ролей. Якщо користувач присутній у списку, йому відкривається доступ до ресурсу, в іншому випадку - доступ закритий. Користувачам, яким відмовлено в доступі, або

запрошуються на повторну реєстрацією, або перенаправляються на WEB-сторінку з повідомленням про закриття доступу.

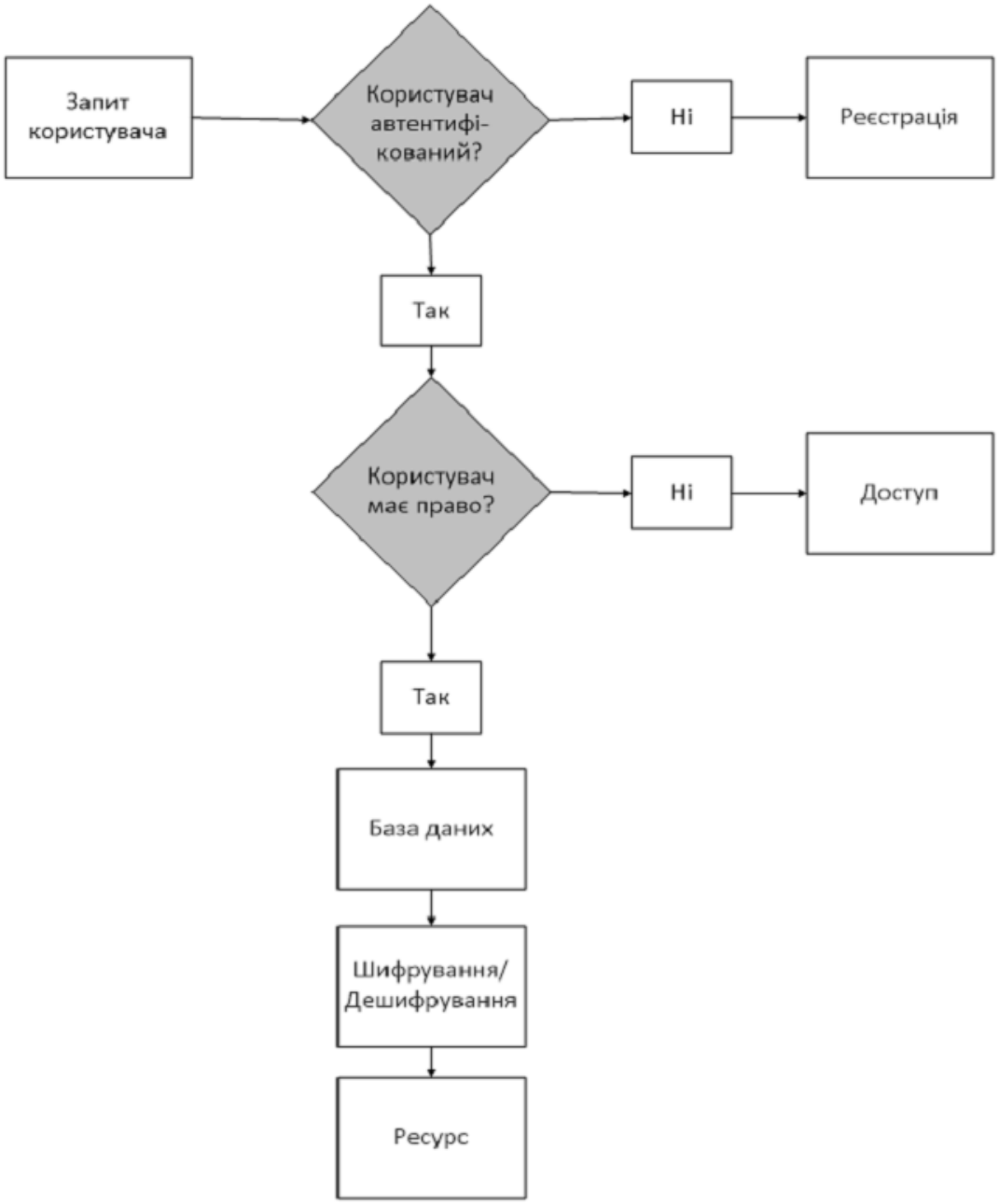

Рисунок 3.4 – Схема алгоритму контролю доступу до веб-додатку

3.5 Шифрування персональних даних користувачів

Хоча автентифікація та авторизація – два найважливіших фактори для побудови безпечних веб-додатків, не слід випускати з уваги і багато інших факторів. Одним з найбільш важливих прикладів є підтримка криптографії – науки шифрування даних для забезпечення конфіденційності і додавання хеш-кодів для визначення цілісності.

За допомогою хешування зберігається контрольна суму вихідних даних, але не самі дані. В результаті немає можливості повернути назад процес хешування для відновлення вихідних даних. Все, що можна зробити – хешувати нові дані і виконати порівняння. Підхід, заснований на хешуванні – найбільш безпечний спосіб перевірки достовірності паролів.

Часто розробники стикаються з цією ситуацією, зберігаючи цінні дані в вигляді відкритого тексту. Вони припускають, що оскільки дані зберігаються в захищеному місці на сервері, їм не потрібно брати на себе додаткову роботу по їх шифруванню. Однак експерти з безпеки знають, що це не так. Без шифрування зловмисникові достатньо отримати доступ до сервера всього на кілька хвилин або навіть секунд, щоб витягти паролі або номери кредитних карток кожного замовника. Проломи в захисті трапляються через непродуману адміністративну політику, слабкі паролі адміністраторів чи вразливого програмного забезпечення сервера. Проблеми можуть виникнути і з причин апаратного характеру; насправді багато компаній продають або викидають старі жорсткі диски серверів без видалення цінних даних, які вони містять. Тому багато організацій дотримуються політики, що вимагає постійного обов'язкового зберігання інформації замовників у конфіденційному зашифрованому вигляді. Якщо виявляється пролом у захисті, компанія буде змушена сповістити користувачів про те, що їхні дані піддалися ризику, тому що вони не були правильно зашифровані. Така компанія зіткнеться з серйозними проблемами і, можливо, повністю втратить довіру замовників. Щоб уникнути цього і гарантувати безпеку даних, потрібно шифрувати важливу інформацію, що зберігається у веб-додатку.

3.6 Висновки до розділу

Однією з найважливіших частина веб-додатку є безпека, вона повинна братися до увагу з першої стадії процесу розробки. Для її забезпечення використовується декілька механізмів, включаючи ідентифікацію користувачів, видачу або позбавлення прав доступу до важливих ресурсів, а також захист інформації, яка зберігається на сервері. Шифрування даних користувачів є принципово важливим показником безпеки веб-додатка. У всіх цих випадках необхідний фундаментальний каркас, що забезпечує базову функціональність безпеки.

У цьому розділі було розглянуто основні методи автентифікації користувачів які використовуються в удосконаленій системі автентифікації та контролю доступом, а також описано алгоритм авторизації користувачів та методику шифрування даних.

РОЗДІЛ 4
УДОСКОНАЛЕНА СИСТЕМА КОМПЛЕКСНОЇ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ І УПРАВЛІННЯ ДОСТУПОМ

4.1 Метод єдиного входу

З метою покращення зручності використання веб-додатку було додано модуль автентифікації за технологією єдиного входу (SSO). На просторах інтернету можна знайти багато провайдерів функціоналу для технології єдиного входу для веб-додатків, проте в даній роботі було використано модуль від одного з найбільш широко-використовуваних провайдерів у світі - okta.

Використання функціоналу для реалізації технології єдиного входу від okta є безкоштовним у користуванні та дозволяє створювати та керувати користувачами, серверами авторизації та безліччю інших завдань, які спрощують обробку веб-автентифікації.

Для початку налаштування сервера авторизації спочатку потрібно створити безкоштовний обліковий запис розробника Okta: https://developer.okta.com/signup/ .

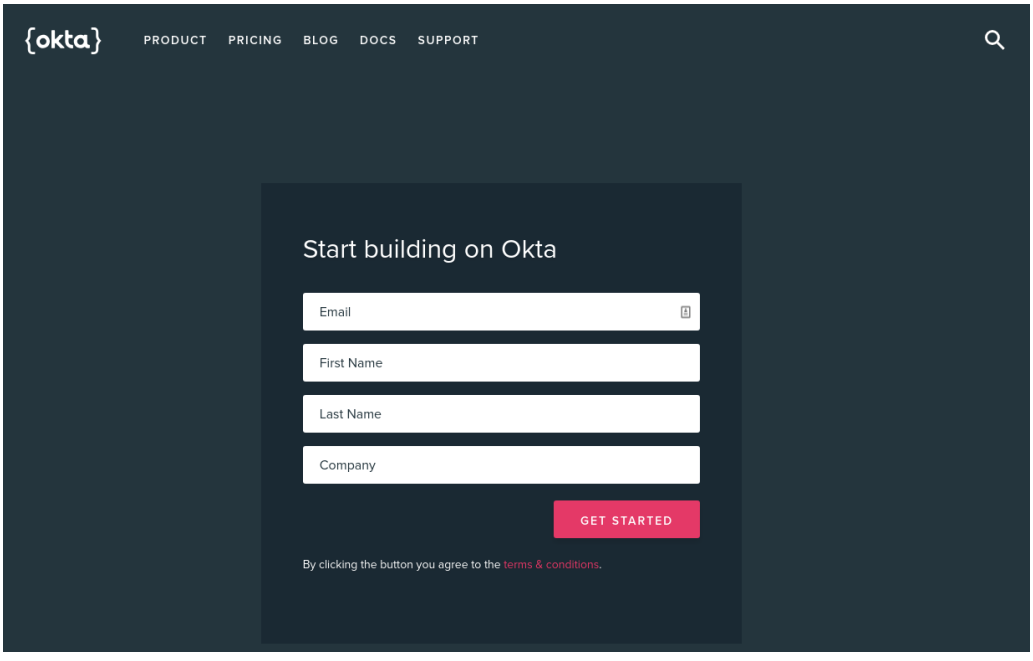

Рисунок 4.1 Створення акаунту в Okta

Після реєстрації необхідно обрати платформу та сконфігурувати параметри автентифікації для неї на панелі управління.

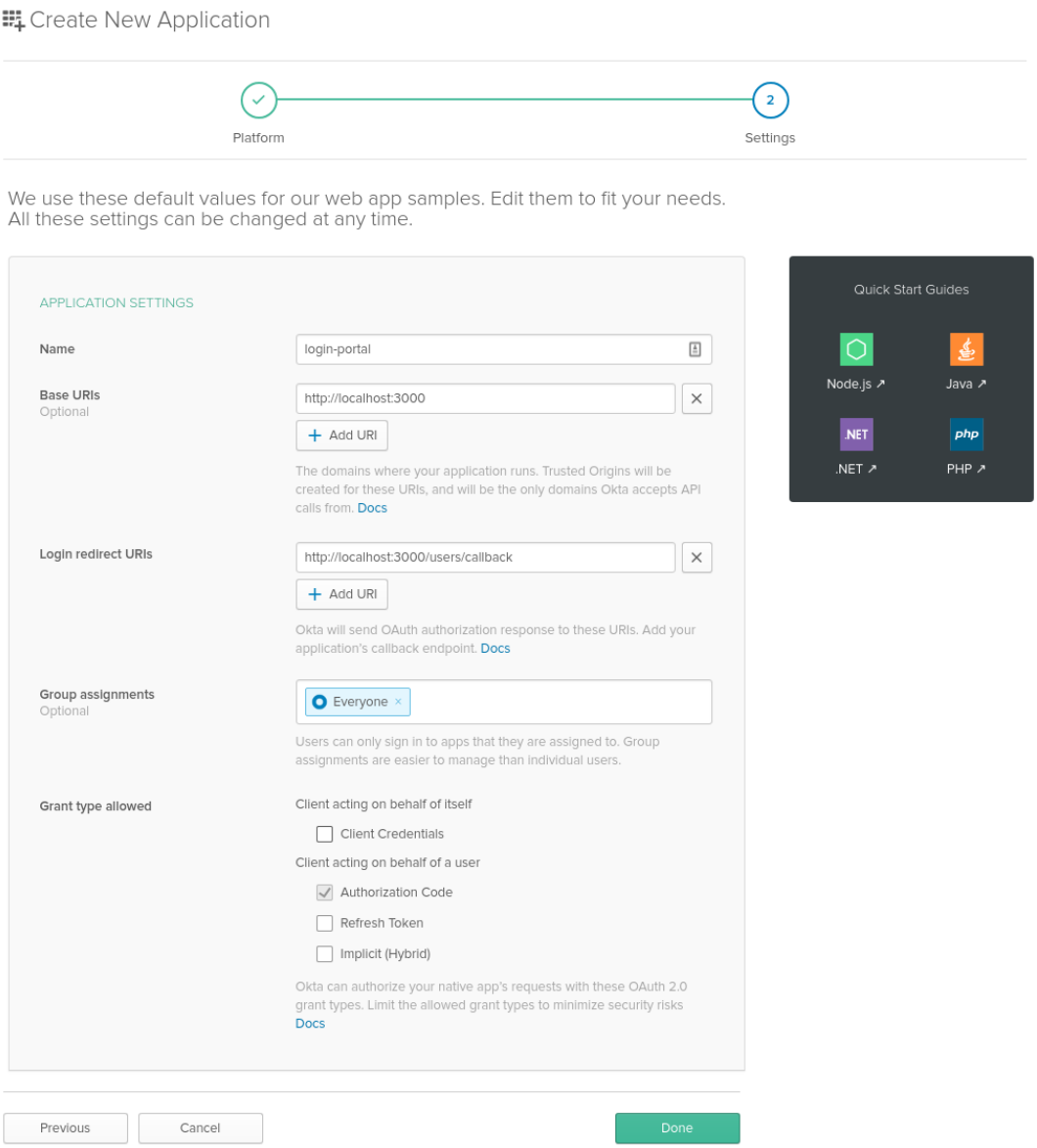

Рисунок 4.2 Налаштування параметрів автентифікації до веб-додатку.

Окрім налаштування okta з панелі управління залишається завдання інтеграції з веб-додатком та налаштування сесій.

Управління сесіями є ядром будь-якої системи автентифікації. Це те, що дозволяє користувачеві залишатися авторизованим на вашому сайті і не потрібно повторно вводити свої облікові дані перед переглядом кожної сторінки. Самий безпечний спосіб для призначених для користувача сеансів -

з допомогою серверних куків. Для керування сеансами було обрано бібліотеку express-session. Ця бібліотека надає можливості:

- Створювати безпечні файли cookie з криптографічним підписом, щоб сервер міг зберігати дані в браузері користувача. Криптографічне підписання - це техніка, яка дозволяє вашому серверу визначити, чи намагався користувач «модифікувати» свої файли cookie.
- Надає простий API для створення та видалення файлів cookie.
- Дозволяє налаштувати параметри cookie на основі того, що потрібно з ними зробити.

Підключення бібліотеки для сесій:

```
app.use(logger('dev'));

app.use(express.json());

app.use(express.urlencoded({ extended: false }));

app.use(cookieParser());

app.use(express.static(path.join(__dirname, 'public')));

app.use(session({

  secret: 'LONG_RANDOM_STRING_HERE',

  resave: true,
```

saveUninitialized: false

}));

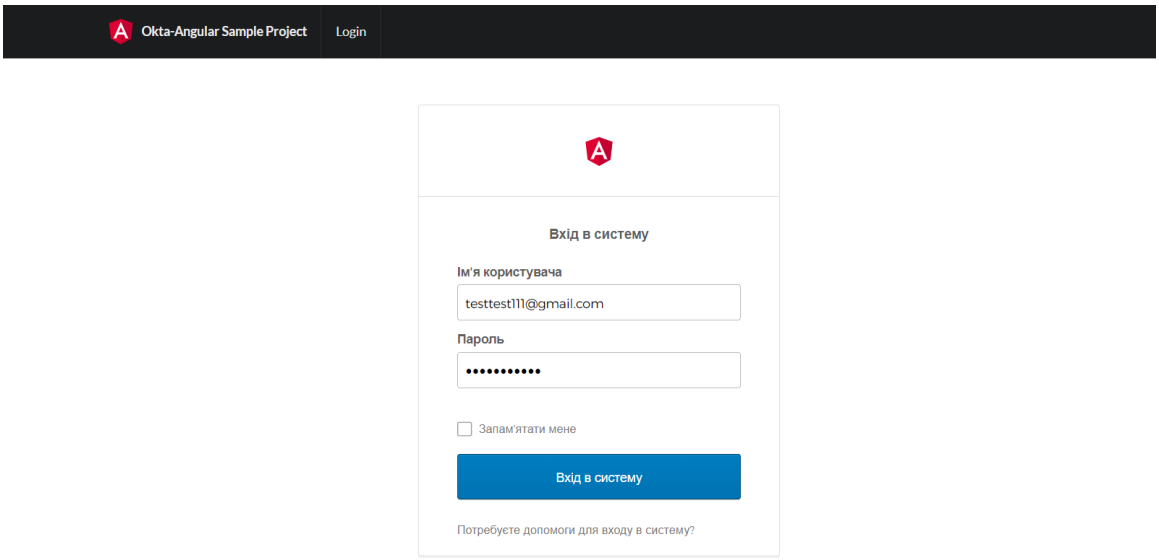

Рисунок 4.3 Вікно єдиного входу до веб-додатку

Список всіх доступних веб-ресурсів для яких діє технологія єдиного входу можна переглянути авторизованим користувачам на панелі доступу до додатків okta:

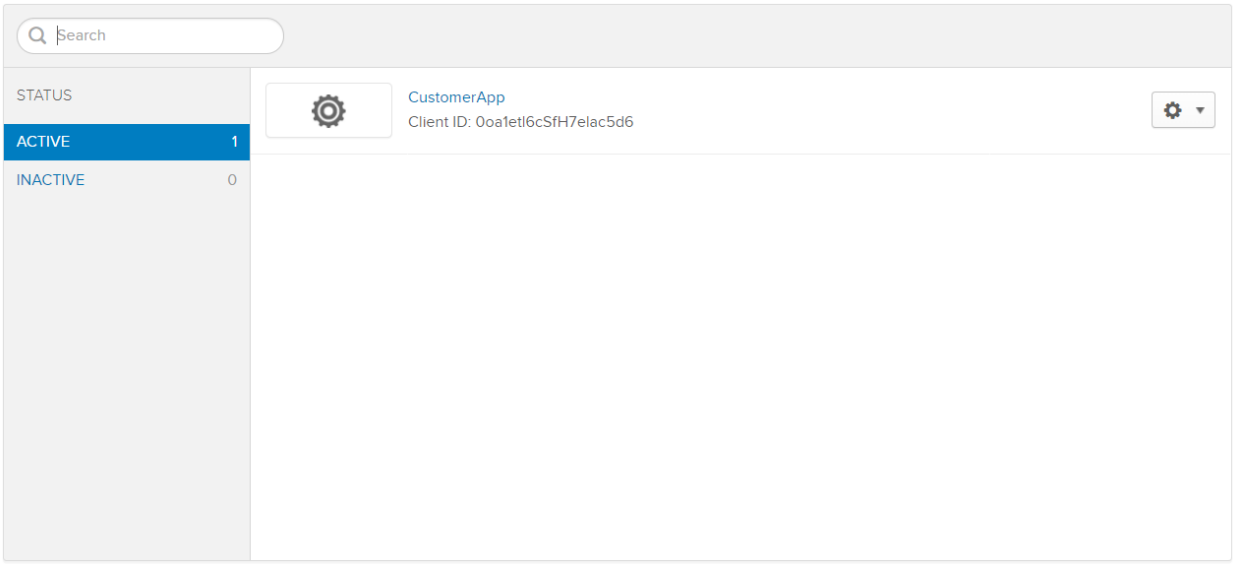

Рисунок 4.4 Вікно доступу до додатків

4.2 Обробка паролів

При обробці потрібно унеможливити збій в програмі, коли користувач ввівши один пароль, отримував доступ до іншого аккаунту, або отримував інші права (вищі ніж його поточні або навіть права адміністратору).

При зберіганні паролів, можна зберігати їх в зашифрованому вигляді. В такому випадку, якщо якимось чином виник витік даних (або виник неавторизований доступ до даних з паролями системи), тоді зловмисник не зможе використати паролі ніяк інакше, як розшифрувавши закодовані дані. Що буде потребувати значних часових та машинних ресурсів (при використанні надійних алгоритмів шифрування).

Функція перевірки даних користувача в системі:

```
UserSchema.statics.authenticate = function (username, password, callback)
{
   User.findOne({ userid: username })
    .exec(function (err, user) {
      if (err) {
        return callback(err)
      } else if (!user) {
        var err = new Error('Користувача не знайдено!');
        err.status = 401;
        return callback(err);
      } else {
      bcrypt.compare(password , user.password, function (err, result) {
        if (result === true) {
```

```
            return callback(null, user);
          } else {
            return callback('Не вірний пароль!');
          }
        })
      }
    });
  }
```

При передачі даних потрібно уникати (і якщо це не дозволяє поточна задача, то мінімізувати) передачу паролів у відкритому вигляді (через GET запит). Намагайтеся передавати паролі завжди POST запитом.

При роботі з банківською та іншою важливою інформацією, бажано користуватися захищеною передачою даних (https) з використанням криптографічних систем (використання SSL). При чому можна шифрувати як паролі до веб-систем, так і всю іншу інформацію.

4.3 Зберігання паролів

Як вже зазначалося, для надійного зберігання паролів, зокрема у файлах, необхідно їх шифрувати. Для цієї задачі найбільш часто використовують хеш-функції. Серверні мови програмування, такі як Perl та PHP, підтримують різноманітні алгоритми шифрування, в тому числі хеш-функції (для проведення хешування), такі як DES, MD4, MD5, SHA1 та інші.

Для надійного та безпечного зберігання пароля в файлі скрипта, або у зовнішньому файлі, потрібно захешувати його функцією хешування (crypt()). Для пароля MbQGWw4s зашифрований crypt() пароль буде виглядати як

Cd46XY120Htc2. При використанні crypt() потрібно враховувати, що функція використовує лише перші 8 символів паролю [8].

При розміщенні зашифрованого пароля в середині скрипта, алгоритм роботи буде наступним:

```
#!/usr/bin/crypt

$enc_password = "Cd46XY120Htc2";

$salt = substr($enc_password,0,2);

# отримуємо пароль від користувача в $password

# та перевіряємо його

if (crypt($password,$salt) eq $enc_password) {

# пароль вірний

}

else {

# пароль невірний

}
```

Якщо зашифрований пароль зберігається в середині скрипта, то отримати доступ до паролю просто так не вийде, бо потрібно буде отримати доступ до вихідних кодів скрипта, щоб прочитати пароль, а для цього потрібно мати доступ до файлової системи. Звичайно можуть бути вразливості на сайті чи у веб сервері, що дозволять прочитати вихідні коди додатку (і відповідно пароль), але при відсутності даних вразливостей (в чому потрібно переконатися), даний варіант зберігання паролів є достатньо надійним.

Якщо зашифрований пароль (чи декілька паролів) зберігається в окремому файлі, то потрібно переконатися, що даний файл з паролями не є доступним для читання з браузеру. Для цього вони повинні мати відповідно налаштовані права (і, при необхідності, розширення файлів), щоб вони були доступні для читання лише для локальних скриптів, але не з браузеру.

В будь-якому разі використання шифрування паролів робить їх зберігання більш безпечним, що покращує безпеку веб додатків у цілому. І навіть при витоку інформації про зашифрований пароль, на його взлом знадобиться дуже багато часу [9].

4.4 Двофакторна автентифікація

Двофакторна автентифікація (ДФА, англ. two-factor authentication, також відома як двоетапна верифікація), є типом багатофакторної автентифікації. ДФА — представляє собою технологію, що забезпечує ідентифікацію користувачів за допомогою комбінації двох різних компонентів.

Хорошим прикладом двофакторної автентифікації є авторизація Google і Microsoft. Коли користувач заходить з нового пристрою, крім автентифікації по імені та паролю, його просять ввести шестизначний (Google) або восьмизначний (Microsoft) код підтвердження. Ви можете отримати його за допомогою SMS, або голосового дзвінка на ваш телефон, він може бути взятий з заздалегідь складеного реєстру разових кодів або ви можете використовувати додаток-аутентификатор, генеруючий новий одноразовий пароль за короткі проміжки часу. Вибрати один з методів можна в налаштуваннях вашого Google або Microsoft-акаунта.

Переваги двофакторної автентифікації:

- Код підтвердження постійно змінюється, а це безпечніше, ніж однофакторний логін-пароль

Недоліки двофакторної автентифікації:

- Відправлений код, може бути перехоплений.
- Повідомлення приходять з деякою затримкою, так як деякий час йде на перевірку.
- Якщо повідомлення приходить на пошту, то лист з кодом може потрапити в спам.

Для забезпечення більшої надійності другого параметру автентифікації, було вирішено використовувати функцію перевірки коду отриманого від серверу автентифікації [11]. Користувач може обрати одну з декількох функцій перевірки у власному кабінеті веб-додатку. Другий параметр автентифікації - це число що відправляється на поштову скриньку користувача, після отримання якого він вираховує результат функції з застосуванням надісланого числа. Результат він вводить у форму другого кроку автентифікації.

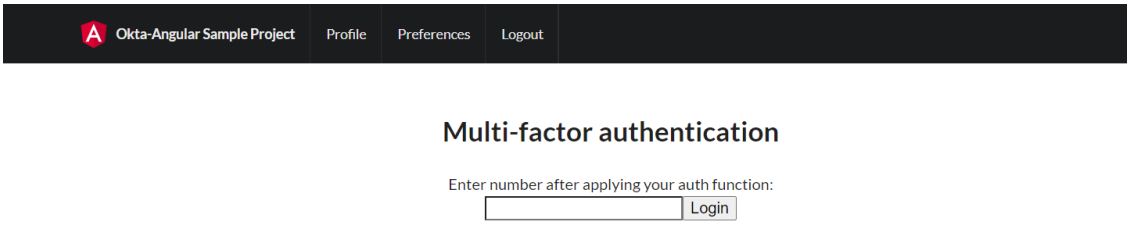

Рисунок 4.4 Форма другого кроку автентифікації

Функцію другого кроку можна вибрати на сторінці налаштувань авторизованого користувача.

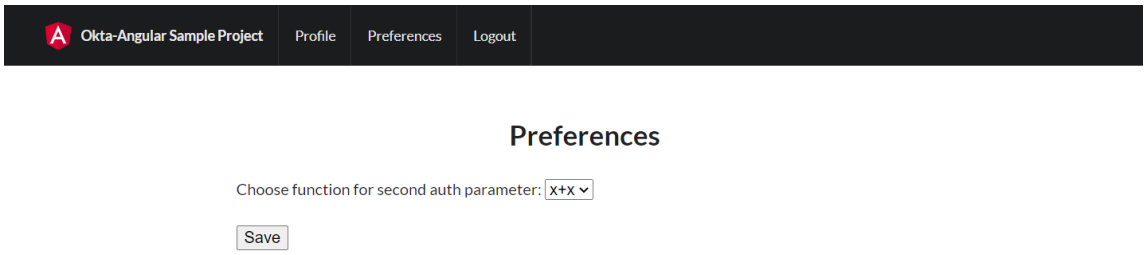

Рисунок 4.5 Сторінка вибору функції перевірки другого параметру автентифікації

4.5 Розмежування прав доступу

Усі веб-системи які надають персоналізований доступу до веб-ресурсу великій кількості користувачів мають систему розмежування прав доступу. І тому, наявність надійної, потужної і водночас зручної системи розмежування прав - це один з головних аспектів, що стосуються інформаційної безпеки вашого веб-сайту.

В більшості веб-додатків розмежування прав відбувається за схемою:

- Адміністратор
- Користувач
- Гість

Сюди ж можуть додаватися різні проміжні групи користувачів, наділені частковими правами. Наприклад така категорія як Модератори. Котрі мають додаткові права по відношенню до звичайних користувачів, але більш обмежені порівняно з адміністраторами. Також можуть бути інші додаткові категорії (VIP-персони, Модератори різного рівня, Забанені користувачі).

Всі URL доступ до яких обмежений роллю користувача системи будуть мати middleware, який перед надсиланням даних до контролеру та виконання певної логіки виконує перевірки прав поточного користувача і якщо користувачу дозволено виконувати дану дію, йому відкриється сторінка, яку він запитував. Якщо у користувача недостатньо прав доступу, йому буде відображена стандартна сторінка із повідомленням, що в нього недостатньо прав для доступу до даної сторінки.

```php
/**
 * users routes
 */
Route::post('/users/filter/{name}', 'UserController@filterStudents')
    ->where(['name' => '[A-Za-z]+'])->middleware('admin');
Route::post('/users/filterbysurname/{sname}', 'UserController@getStudentsBySurname')
    ->where(['name' => '[A-Za-z]+'])->middleware('admin');
Route::post('/users/delete/{id}','UserController@destroy')->middleware('admin');
/**
 * groups routes
 */
Route::group(['middleware' => ['auth','admin'], 'prefix' => 'groups'], function (){

    Route::get('/', 'GroupController@index')->name('groups');

    Route::get('/data',  'GroupController@getGroupsData');

    Route::post('/add', 'GroupController@store');

    Route::put('/update/{groupId}', 'GroupController@update')
        ->where(['groupId' => '[0-9]*']);

    Route::post('/filter', 'GroupController@filterGroups');

    Route::delete('/delete/{id}','GroupController@destroy');

    Route::post('/get-teacher-by-course-and-direction','GroupController@getTeacher');

    Route::post('{id}/restore','GroupController@restore');
});
```

З даного фрагменту можна помітити, що для задання middleware, потрібно передати асоціативний масив, де ключем масиву буде виступати власне назва цього помічника, а параметрами виступає масив, в який передається метод, який повинен бути викликаний для перевірки прав користувача на виконання даних дій. Наступним параметром масиву передається prefix, який забезпечує додавання до всіх роутів даного введеного розробником слова. Тобто, якщо розглянути перший роут для груп, то він буде виглядати наступний чином – groups/. Кожний роут системи складається з певних частин та відповідає за надсилання даних до конкретного контролера системи. На початку кожного правила йде зарезервоване ключове слово Route. Далі через двокрапку вказується назва HTTP методу, який бути задіяний.

Таблиця 4.1 Основні HTTP-методи

Назва методу	Дія, яку повинен виконувати
POST	Створення певного нового запису
GET	Читання певних даних без внесення змін до них. Використовується, наприклад, для отримання контенту головної сторінки
PUT	Оновлення та повна заміна даних
PATCH	Оновлення та зміна певного поля для певної таблиці
DELETE	Видалення даних

Після того, як визначено метод, який повинен використовуватися в залежності від виконуваних дій, прописується URL, при введенні в командному рядку браузера якого, відбувається виклик визначеного контролера. Наступним параметром власне передається назва контролера та

назва конкретного методу, який повинен викликатися при введенні в браузері даного конкретного роута. Додатковий метод name із передачею в нього назви, використовується для того, щоб можна були динамічно формувати на сторінці, яка відображається користувачу правильну URL адресу, вказавши всього лише назву роуту, для якого ми хочемо її сформувати. При натисненні користувача на дану адресу, буде автоматично виклика команда(роут), яка має дане ім’я. При використанні зарезервованого слова resource ситуація виглядить іншим чином. Справа в тому, що дане слово передбачає наявність в контролері семи методів для роботи з певними даними. Дані методи є однаковими для кожного контролера і описують всі дії, які потрібні для роботи з певним типом матеріалів. Це є досить зручним та скорочує кількість програмного коду, оскільки дозволяє одним словом задати сім методів. Також в даних правилах можна вказувати, перелік методів, які повинні використовуватися за допомогою ключового слова only або вказувати методи, які не потрібно використовувати за допомогою слова expect.

Весь процес автентифікації привілейованих користувачів (адміністраторів) можна описати наступною блок-схемою:

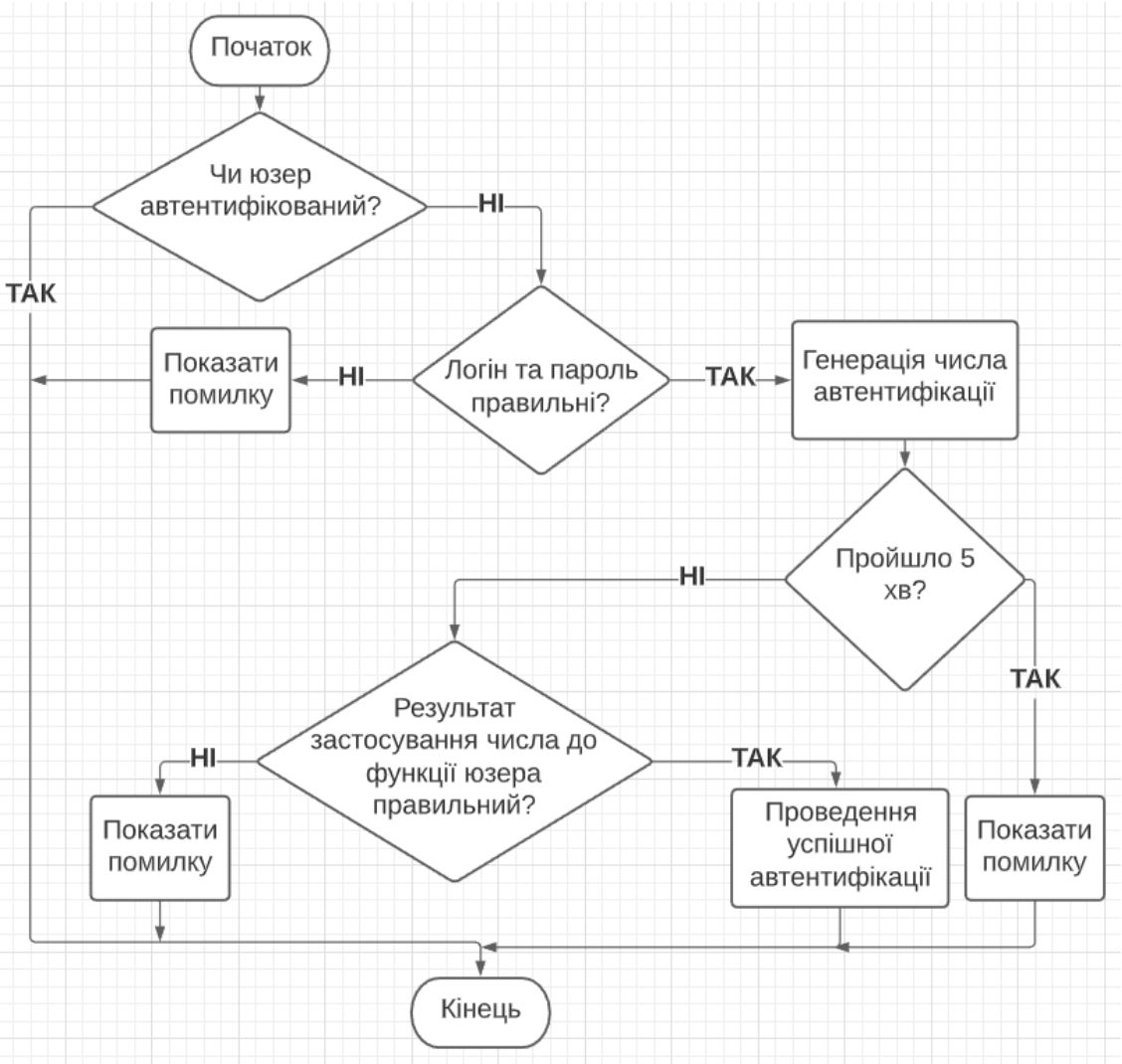

Рисунок 4.6 Блок-схема процесу автентифікації привілейованих користувачів до веб-додатку

4.6 Тестування

Розроблену систему було покрито unit- та інтеграційними тестами, де перевірялись наступні сценарії:

- Перевірка заповнених обов'язкових полів
- Введення коректного логіна і коректного пароля.
- Введення коректного логіна і некоректного пароля

- Введення некоректного логіна і коректного пароля
- Введення некоректного логіна і некоректного пароля
- Перевірка застосування числа до функції автентифікації другого кроку
- Спроба входу на інший ресурс з однієї групи для вже автентифікованого користувача через автентифікацію єдиного входу

Приклад тесту який перевіряє автентифікацію звичайного користувача:

```
describe('Login Page', function () {
  it('should let you log in', function () {
    browser.url('/login');
    browser.setValue('input[name="email"]', 'valid@user.com');
    browser.setValue('input[name="password"]', '1234');
    browser.click('.button=Login');

    const pageUrl = browser.getUrl();
    assert.notEqual(pageUrl, 'http://localhost:8080/login');
    assert.equal(pageUrl, 'http://localhost:8080/home');
  });
});
```

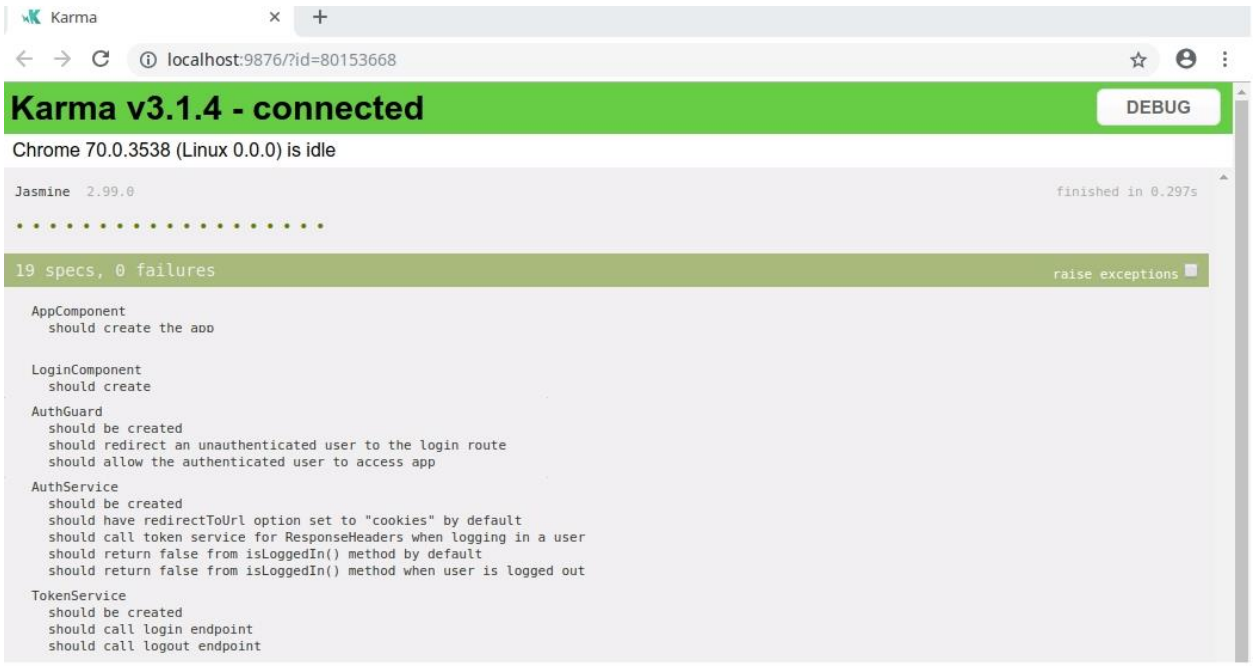

Рисунок 4.7 Результати виконання тестів

4.7 Висновки до розділу

Сучасні веб-додатки можуть виконувати безліч функцій, пов’язаних із забезпеченням бізнес-процесів, адміністрування, модифікацією персональних даних тощо. Забезпечення безпечної, безперебійної і ефективної роботи додатку є однією з найважливіших задач яка значною мірою залежить від процесів автентифікації та контролю доступу.

Проаналізувавши основні характеристики систем автентифікації було обрано рішення, які є найбільш надійними, зручними у використанні та не дорогими у реалізації.

Розроблена система має різні рівні захищеності для різних типів користувачів, а також відзначається надійністю та зручністю у використанні завдяки методу єдиного входу.

ВИСНОВКИ

Інтернет став невід'ємною частиною життя і мережею, що постійно розвивається і яка змінила вид діяльності багатьох людей та організацій. Багато установ були атаковані, що призвело до великих втрат. Мережі організації, що не знають або ігнорують ці проблеми піддають себе великому ризику бути атакованими зловмисниками. Захист інформації, основним завданням якого є запобігання порушенню функцій інформаційної безпеки, займає перше місце по актуальності з поставлених задач.

В даній роботі було виконано наступні завдання:

- досліджено сучасні методи автентифікації та керування доступом, що дало змогу зробити висновки про доцільну архітектуру розроблюваної системи;
- розроблено алгоритми автентифікації користувачів різних типів до веб-додатку, а також алгоритми розмежування доступу, що забезпечує підвищений захист процесу автентифікації для привілейованих користувачів, а також контроль доступу до веб-сторінок.
- програмно реалізовано систему автентифікації та контролю доступу, яка забезпечує надійність, безпеку та зручність для її користувачів на основі розроблених алгоритмів, що забезпечує безпеку процесу автентифікації та зручність у користуванні у веб-додатком за рахунок технології єдиного входу, додаткових перевірок двофакторної автентифікації та перевірки доступу до адрес сторінок веб-додатку.

Дана система була спроектована на основі аналізу вразливостей та методів автентифікації до веб-додатків. Можна виділити наступні переваги розробленої системи:

- підвищений захист автентифікації привілейованих користувачів;
- зручність у користуванні для адміністраторів, що мають доступ до декількох веб-додатків водночас;
- удосконалений механізм двофакторної перевірки користувачів;

- гнучкість та можливість до розширення новими модулями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Moffett J. Security & Distributed Systems [Електронний ресурс] / Jonathan Moffett – http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.22.2312&rep=rep1&type=pdf.
2. Угрозы несанкционированного доступа [Електронний ресурс]. – 2008. – https://zakonbase.ru/content/part/1183301
3. Martin Fowler. Microservices [Електронний ресурс] — https://martinfowler.com/articles/microservices.html Дата доступу: 09.10.2018
4. Ворожко В.П., Корченко О.Г. Захист інформації з обмеженим доступом. Збірник нормативних документів. – К.: КУЦА, 1999. – 283 с
5. Домарев В.В. Безопасность ИТ. Методология создания систем защиты. – М. – СПб. – Киев, 2002. – 688 с.
6. ISO/IEC 27005:2005 «Інформаційні технологій. Методи захисту. Система управління інформаційною безпекою. Вимоги» (Електрон. Ресурс)/Спосіб доступу: URL:http://www.dstszi.gov.ua/dstszi/control.– Загол. з екрана.
7. Грушо А.А., Тимонина Е.Е. Теоретические основы защиты информации. – М.: Яхтсмен, 1996. – 192 с.
8. Гайкович В.Ю., Ершов Д.В. Основы безопасности информационных технологий. М.: МИФИ, 1995. – 285с.
9. Самохвалов Ю.Я., Темпіков В.О., Хорошко В.О. Організаційно-технічне забезпечення захисту інформації: Навчальний посібник / За ред. проф. Хорошка В.О. – К.: НАУ, 2002. – С. 207
10. С.Н. Ардатский, О.С. Бартунов Управление доступом в сложных информационных системах. – Образовательные порталы России. Выпуск 1. - 2005.-187 с.
11. НД ТЗІ 3.7-001-99 «Методичні вказівки щодо розробки технічного завдання

на створення комплексної системи захисту інформації в автоматизованій системі» // Урядовий кур'єр. – 1999. – № 113. – С. 18.

12. Корченко О. Метод автентифікації користувачів інформаційних систем за їх рукописним почерком з багатокроковою корекцією первинних даних / О. Корченко, А. Давиденко, О. Висоцька // Захист інформації. 2019. – Том 21, №1. – С. 40-51. DOI: 10.18372/2410-7840.21.135462.
13. Висоцька О.О. Моніторинг роботи користувачів комп'ютерних систем за допомогою технологій розпізнавання за клавіатурним почерком / О.О. Висоцька// Моделювання та інформаційні технології. Збірник наукових праць інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН Україні. – К.:ІПМЕ, 2018. – Вип. 84. - С. 119-125.
14. Vysotska O. Keystroke Pattern Authentication of Computer Systems Users as One of the Steps of Multifactor Authentication / O. Vysotska, A. Davydenko // Advances in Intelligent Systems and Computing. 2019. – V. 938.–P. 356-368. DOI: https://doi.org/10.1007/978-3-030-16621-2_33
15. Overview of data loss prevention policies. [Electronic resource] – Acess : https://support.office.com/en-us/article/Overview-of-data-loss-preventionpolicies-1966b2a7-d1e2-4d92-ab61-42efbb137f5e?ui=en-US&rs=enUS&ad=US.
16. Защита от спама в Office 365. [Електронний ресурс] – Режим доступу : https://support.office.com/ru-ru/article/Защита-от-спама-в-Office-365-6a601501-a6a8-4559-b2e7-56b59c96a586
17. Security and Compliance: Customer Controls for Information Protection in Office 365. [Електронний ресурс] – Режим доступу : http://download.microsoft.com/download/F/2/B/F2B9D8BB-30C3-427C8FBE-E687D986BD91/Whitepaper%20-20Customer%20controls%20for%20Information%20protection%20in%20Office%20365.docx
18. НД ТЗІ 1.1-002-99 Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу.

19. ISO/IEC 15408-1:2005 Common Criteria for IT Security Evaluation.
20. Закон України «Про електронні довірчі послуги» від 14.01.2020.
21. Положення про технічний захист інформації в Україні, затверджене Указом Президента України від 27 вересня 1999 р. № 1229 (зі змінами та доповненнями).
22. Концепція інформаційної безпеки України [Електронний ресурс]. – URL: http://mip.gov.ua/done_img/d/30-project_08_06_15.pdf