МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ І ПРИКЛАДНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЛУПОЛ АНТОН АНАТОЛІЙОВИЧ

допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій,
д-р. техн. наук, професор
_____Наталія ВЕСЕЛОВСЬКА
«____»_____ 2026 р.

МЕТОДИ ІЗОЛЯЦІЇ БРАУЗЕРНИХ ПРОФІЛІВ ТА КЕРУВАННЯ ПАРАМЕТРАМИ ЦИФРОВОЇ ІДЕНТИФІКАЦІЇ У ВЕБСЕРЕДОВИЩІ

Спеціальність 122 «Комп’ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:
Юрій АНТОНОВ, кан. фіз.-мат. наук, доцент,
доцент кафедри інформаційних
технологій

_____
(Підпис)

Оцінка _____/_____/_____
(бали/за шкалою ЄКТС/за національною шкалою)
Голова ЕК: _____
(Підпис)

Вінниця 2026

АНОТАЦІЯ

Лупол А.А. Методи ізоляції браузерних профілів та керування параметрами цифрової ідентифікації у веб-середовищі. Спеціальність 122 «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2026.

У роботі досліджено методи цифрової ідентифікації користувачів та механізми браузерного фінгерпринтингу. Розроблено програмний засіб IdentityLab для створення ізольованих браузерних профілів, керування параметрами цифрової ідентифікації та аналізу браузерного відбитка. Реалізовано механізми User-Agent, Navigator, Timezone і WebGL Spoofing, а також модулі Fingerprint Analyzer та Risk Score. Проведене тестування підтвердило працездатність розробленого програмного засобу.

Ключові слова: цифрова ідентифікація, браузерний профіль, браузерний фінгерпринтинг, Electron, Chromium, User-Agent, WebGL, ізоляція профілів.

54 с., 6 рис., 1 табл., 42 джерела.

ABSTRACT

Lupol A.A. Methods of Browser Profile Isolation and Management of Digital Identity Parameters in Web Environment. Specialty 122 "Computer Science". Vasyl Stus Donetsk National University, Vinnytsia, 2026.

The paper investigates digital identity mechanisms and browser fingerprinting techniques. The IdentityLab software system was developed for creating isolated browser profiles, managing digital identity parameters, and analyzing browser fingerprints. The system implements User-Agent, Navigator, Timezone, and WebGL Spoofing mechanisms, as well as Fingerprint Analyzer and Risk Score modules. Testing confirmed the correctness and operability of the developed software.

Keywords: digital identity, browser profile, browser fingerprinting, Electron, Chromium, User-Agent, WebGL, profile isolation.

54 p., 6 fig., 1 table, 42 sources.

ЗМІСТ

ВСТУП

Сучасний розвиток вебтехнологій супроводжується активним використанням механізмів цифрової ідентифікації користувачів. Під час взаємодії з вебресурсами браузер передає значну кількість характеристик програмного та апаратного середовища, які можуть використовуватися для формування унікального цифрового відбитка пристрою. На відміну від традиційних механізмів відстеження, що базуються на використанні файлів cookie, сучасні методи браузерного фінгерпринтингу дозволяють ідентифікувати користувача на основі сукупності параметрів браузера, операційної системи, графічної підсистеми, мовних налаштувань та інших характеристик веб-середовища.

Технології цифрової ідентифікації широко використовуються в електронній комерції, системах вебаналітики, сервісах захисту від шахрайства, рекламних платформах та інших інформаційних системах. Водночас їх використання породжує низку питань, пов'язаних із конфіденційністю користувачів, контролем персональних даних та можливістю керування параметрами цифрового відбитка.

Одним із напрямів дослідження даної проблематики є створення ізольованих браузерних профілів із можливістю зміни окремих параметрів цифрової ідентифікації. Такі рішення дозволяють досліджувати механізми формування браузерних відбитків, аналізувати вплив різних характеристик на процес ідентифікації та створювати незалежні браузерні середовища для виконання спеціалізованих завдань.

Актуальність теми обумовлена зростанням ролі технологій браузерного фінгерпринтингу у сучасному вебпросторі та необхідністю дослідження методів ізоляції браузерних профілів і керування параметрами цифрової ідентифікації. Створення програмних засобів для формування контрольованого браузерного середовища дозволяє більш детально аналізувати процеси цифрової ідентифікації та оцінювати вплив окремих характеристик браузера на формування цифрового відбитка.

Метою роботи є розробка програмного засобу для ізоляції браузерних профілів та керування параметрами цифрової ідентифікації у вебсередовищі.

Для досягнення поставленої мети необхідно було вирішити такі завдання:

- дослідити принципи формування цифрового відбитка браузера;
- проаналізувати сучасні методи браузерної ідентифікації користувачів;
- виконати огляд існуючих засобів керування браузерними профілями;
- спроєктувати архітектуру програмного засобу;
- реалізувати механізми ізоляції браузерних профілів;
- реалізувати механізми керування параметрами цифрової ідентифікації;
- реалізувати засоби аналізу браузерного відбитка;
- розробити механізм оцінювання доступності fingerprint-сигналів;
- провести тестування розробленого програмного засобу та оцінити результати його роботи.

Об'єктом дослідження є процеси цифрової ідентифікації користувачів у вебсередовищі.

Предметом дослідження є методи ізоляції браузерних профілів та керування параметрами цифрової ідентифікації під час взаємодії з вебресурсами.

Під час виконання роботи використовувалися такі методи дослідження: аналіз наукових і технічних джерел, системний аналіз, проєктування програмного забезпечення, моделювання структури браузерних профілів, експериментальне дослідження та тестування програмного забезпечення.

Практичне значення отриманих результатів полягає у розробці програмного засобу IdentityLab, який забезпечує створення ізольованих браузерних профілів, керування параметрами цифрової ідентифікації та аналіз характеристик браузерного відбитка. Результати роботи можуть бути використані для дослідження механізмів браузерного фінгерпринтингу, тестування вебресурсів та вивчення особливостей цифрової ідентифікації у вебсередовищі.

Структурно бакалаврська робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. У першому розділі виконано

аналіз предметної області та сучасних методів цифрової ідентифікації користувачів. У другому розділі розглянуто питання проєктування програмного засобу та архітектурних рішень. Третій розділ присвячено реалізації системи IdentityLab та результатам її тестування.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ЦИФРОВОЇ ІДЕНТИФІКАЦІЇ У ВЕБ-СЕРЕДОВИЩІ

1.1 Поняття цифрової ідентифікації користувача у веб-середовищі.

Стрімкий розвиток вебтехнологій призвів до того, що більшість повсякденних дій користувачів виконується через браузер. Онлайн-банкінг, електронна комерція, соціальні мережі, освітні платформи та інші вебресурси постійно взаємодіють із користувачем і потребують механізмів його ідентифікації. Для забезпечення персоналізації сервісів, підтримки безпеки та захисту від шахрайських дій вебсайти збирають і аналізують різноманітні дані про пристрій та програмне середовище користувача. Сукупність таких даних формує цифрову ідентифікацію користувача у веб-середовищі. [1, 5]

Під цифровою ідентифікацією розуміють набір характеристик, які дозволяють вебресурсам розпізнавати користувача або його пристрій під час взаємодії з мережею Інтернет. На відміну від традиційної ідентифікації, яка базується на логіні, паролі чи інших облікових даних, цифрова ідентифікація може здійснюватися навіть без прямого введення користувачем будь-якої інформації. Достатньо самого факту відвідування вебсторінки, щоб сайт отримав певний набір технічних параметрів браузера та операційної системи. [1–5]

Одним із найпоширеніших механізмів ідентифікації тривалий час залишалися файли cookie. Вони являють собою невеликі текстові дані, які вебсайт зберігає у браузері користувача для подальшого розпізнавання під час наступних відвідувань. За допомогою cookie забезпечується авторизація, збереження налаштувань користувача, підтримка кошиків покупок в інтернет-магазинах та інші функції. Разом із тим поширення політик конфіденційності та обмеження використання сторонніх cookie призвели до активного розвитку альтернативних методів відстеження користувачів. [12, 38, 40]

Сучасні веббраузери надають сайтам доступ до великої кількості технічних характеристик через програмні інтерфейси JavaScript. Серед них можна виділити відомості про операційну систему, тип і версію браузера, мову

інтерфейсу, часовий пояс, роздільну здатність екрана, кількість логічних процесорів, обсяг оперативної пам’яті та інші параметри. Кожна окрема характеристика не є унікальною, однак їх сукупність може дозволити достатньо точно ідентифікувати конкретний пристрій серед великої кількості інших. [6, 7, 10, 11]

Важливу роль у процесі цифрової ідентифікації відіграє браузерний профіль. Під браузерним профілем доцільно розуміти набір налаштувань, локальних даних та характеристик середовища виконання, які використовуються браузером під час роботи. До складу профілю можуть входити файли cookie, кеш, локальні сховища даних, історія переглядів, параметри відображення вебсторінок, мовні налаштування та інші дані. Саме браузерний профіль у багатьох випадках виступає основним джерелом інформації для формування цифрового образу користувача. [12, 13, 15]

З розвитком технологій аналізу поведінки користувачів з’явився окремий напрямок досліджень, пов’язаний із браузерним фінгерпринтингом. Його метою є формування так званого цифрового відбитка браузера на основі набору доступних технічних параметрів. На відміну від cookie, цифровий відбиток не потребує збереження даних на пристрої користувача, оскільки генерується безпосередньо під час відвідування вебсторінки. Це робить його більш стійким до очищення локальних даних або використання режимів приватного перегляду. [1–5]

Питання цифрової ідентифікації набуває особливої актуальності у випадках, коли користувачеві необхідно працювати з кількома незалежними обліковими записами або забезпечити розділення даних між різними видами діяльності. У таких ситуаціях виникає потреба в механізмах ізоляції браузерних профілів та контролю параметрів, які можуть використовуватися для ідентифікації пристрою. Реалізація таких механізмів дозволяє створювати незалежні середовища роботи браузера та керувати набором характеристик, що передаються вебресурсам під час взаємодії.

Таким чином, цифрова ідентифікація є невід’ємною складовою сучасного веб=середовища та використовується для персоналізації сервісів, забезпечення безпеки й аналізу поведінки користувачів. Водночас широкі можливості збору технічних параметрів браузера створюють передумови для формування детального цифрового профілю користувача. Це обумовлює необхідність дослідження методів ізоляції браузерних профілів і керування параметрами цифрової ідентифікації, які розглядаються в подальших розділах роботи.

1.2 Методи формування браузерного відбитку

Під час взаємодії користувача з вебресурсами браузер постійно обмінюється інформацією із сервером та надає доступ до різноманітних програмних інтерфейсів. Частина цих даних необхідна для коректного відображення вебсторінок та забезпечення сумісності між різними пристроями. Водночас сукупність таких характеристик може використовуватися для формування браузерного відбитка (Browser Fingerprint), який дозволяє ідентифікувати окремий браузер або пристрій серед великої кількості інших. [1–5]

Браузерний відбиток являє собою набір технічних параметрів, отриманих під час роботи браузера. На відміну від файлів cookie, які зберігаються локально на пристрої користувача, цифровий відбиток формується динамічно під час відвідування вебсторінки. Це робить його більш стійким до очищення локальних даних та використання режимів приватного перегляду. [1–5]

Одним із базових джерел інформації є рядок User-Agent. Він передається браузером у складі HTTP-запиту та містить відомості про браузер, його версію, операційну систему та платформу. Наприклад, сучасний браузер Google Chrome на операційній системі Windows передає інший User-Agent, ніж Safari на macOS. Вебресурси використовують цю інформацію для адаптації інтерфейсу, однак вона також може бути використана для побудови цифрового профілю користувача. [6, 14]

Значний обсяг даних доступний через об’єкт Navigator, який входить до складу браузерного середовища JavaScript. За його допомогою вебсторінка може

отримати відомості про операційну систему, підтримувані мови, апаратні характеристики пристрою та інші параметри. [7]

Найбільш поширеними є такі властивості:

- navigator.platform — платформа та архітектура пристрою;
- navigator.language — основна мова браузера;
- navigator.languages — список доступних мов;
- navigator.hardwareConcurrency — кількість логічних процесорів;
- navigator.deviceMemory — обсяг оперативної пам’яті;
- navigator.maxTouchPoints — підтримка сенсорного введення.

Кожен із зазначених параметрів окремо має невелику цінність для ідентифікації, однак їх поєднання дозволяє суттєво звузити коло можливих пристроїв. [7]

Ще одним джерелом інформації виступають параметри екрана користувача. За допомогою об’єкта Screen вебресурси можуть визначати роздільну здатність дисплея, доступну робочу область, глибину кольору та інші характеристики. Додатково браузер надає доступ до значення devicePixelRatio, яке характеризує співвідношення між фізичними та логічними пікселями. Особливо корисним цей параметр є для визначення дисплеїв із високою щільністю пікселів, зокрема Retina-дисплеїв у пристроях Apple. [11]

Важливу роль у процесі формування браузерного відбитка відіграють регіональні налаштування. Вебсторінка може отримати інформацію про часовий пояс користувача, локалізацію браузера та мовні параметри системи. Наприклад, поєднання значень Europe/Kyiv, uk-UA та української локалізації значно звужує множину потенційних конфігурацій. Саме тому часовий пояс та мовні налаштування часто враховуються під час аналізу цифрової ідентифікації. [6, 7]

Одним із найвідоміших сучасних методів фінгерпринтингу є використання технології Canvas. Вона дозволяє браузеру створювати графічні зображення за допомогою HTML5 Canvas API. Незважаючи на однаковий програмний код, результат рендерингу може дещо відрізнятися залежно від операційної системи, драйверів, відеокарти та використовуваних шрифтів. Аналіз отриманого зображення дозволяє сформувати додатковий ідентифікаційний параметр. [8]

Схожий принцип використовується і для WebGL Fingerprinting. Технологія WebGL призначена для відображення тривимірної графіки у браузері та забезпечує доступ до інформації про графічний адаптер пристрою. За допомогою спеціальних запитів вебсторінка може отримати відомості про виробника графічного процесора та використовуваний механізм рендерингу. Наприклад, різні пристрої можуть повертати значення, що відповідають Intel Iris Graphics, NVIDIA GeForce, AMD Radeon або графічним рішенням Apple. Оскільки такі характеристики часто мають високу унікальність, WebGL вважається одним із найбільш інформативних джерел даних для побудови цифрового відбитка. [9, 17, 18]

Окрім графічних технологій, сучасні системи аналізу можуть використовувати й інші джерела інформації. До них належать AudioContext Fingerprinting, аналіз встановлених шрифтів, підтримуваних браузером API, особливостей роботи JavaScript-рушія та мережевих характеристик. Кожен із цих сигналів окремо не забезпечує однозначної ідентифікації, проте їх комбіноване використання дозволяє значно підвищити точність розпізнавання браузера. [1–5]

Слід зазначити, що сам факт доступності певного параметра не означає можливість точного встановлення особи користувача. Більшість сучасних механізмів фінгерпринтингу працює на основі накопичення великої кількості характеристик і побудови статистичного профілю. Чим більше даних доступно вебресурсу, тим вища ймовірність формування унікального цифрового відбитка. [5]

Таким чином, браузерний відбиток формується на основі сукупності програмних, апаратних та мережевих характеристик пристрою. Найбільш значущими джерелами інформації є User-Agent, параметри Navigator API, характеристики екрана, регіональні налаштування та дані графічної підсистеми. Розуміння принципів формування браузерного відбитка є необхідною передумовою для розробки механізмів ізоляції браузерних профілів та керування параметрами цифрової ідентифікації.

1.3 Методи ізоляції браузерних профілів

Зі збільшенням кількості вебсервісів та онлайн-платформ користувачі все частіше стикаються з необхідністю розділення різних видів діяльності в мережі Інтернет. Наприклад, окремі профілі можуть використовуватися для особистої роботи, навчання, тестування вебдодатків або керування кількома обліковими записами на одному сервісі. У таких випадках виникає потреба забезпечити незалежність даних між різними середовищами браузера та мінімізувати можливість їх взаємного впливу. [12, 13]

Під ізоляцією браузерного профілю розуміють сукупність методів і механізмів, які забезпечують відокремлене зберігання даних та налаштувань браузера для різних користувацьких середовищ. Основною метою ізоляції є запобігання спільному використанню локальних даних між профілями та створення незалежних середовищ виконання вебдодатків. [12, 13, 15]

Найпростішим способом реалізації ізоляції є використання декількох профілів у межах одного браузера. Більшість сучасних браузерів підтримують створення окремих профілів користувача, кожен з яких має власні налаштування, історію переглядів, закладки та файли cookie. Такий підхід дозволяє розділяти особисту та робочу діяльність, проте рівень контролю над параметрами цифрової ідентифікації залишається обмеженим. [25]

Більш глибокий рівень ізоляції забезпечується за рахунок використання окремих директорій зберігання даних. У цьому випадку для кожного профілю створюється власне сховище, в якому розміщуються файли cookie, кеш браузера, локальні бази даних та інші дані, що накопичуються під час роботи. Завдяки цьому інформація, створена в одному профілі, не використовується іншими профілями навіть за умови запуску в межах одного браузерного рушія. [12, 13, 15]

Особливе значення для ізоляції мають сучасні механізми локального зберігання даних, які використовуються вебдодатками. До них належать LocalStorage, SessionStorage та IndexedDB. Багато вебресурсів використовують ці сховища для збереження інформації про користувача, налаштувань

інтерфейсу, токенів автентифікації та інших службових даних. Якщо декілька профілів використовують спільне сховище, це може призвести до небажаного змішування даних або втрати незалежності між обліковими записами. [12, 13, 15]

Одним із найбільш ефективних підходів до ізоляції є використання окремих браузерних сесій. У цьому випадку кожна сесія функціонує як незалежне середовище виконання та має власний набір локальних даних. Для вебресурсів такі середовища виглядають як окремі браузери, навіть якщо фактично вони працюють у межах одного програмного застосунку. Подібний підхід широко використовується в інструментах тестування вебдодатків, системах автоматизації та спеціалізованих браузерах для роботи з кількома обліковими записами. [20, 22, 25]

Окрім локальних даних, ізоляція може стосуватися й параметрів цифрової ідентифікації. У такому випадку кожний профіль отримує власний набір характеристик, які використовуються під час взаємодії з вебресурсами. Це можуть бути різні мовні налаштування, часові пояси, параметри операційної системи або характеристики графічної підсистеми. Поєднання ізольованого сховища та незалежного набору параметрів дозволяє створювати профілі, які не лише зберігають дані окремо один від одного, а й відрізняються з точки зору цифрового відбитка. [1–5]

Окремої уваги заслуговують спеціалізовані програмні засоби, які реалізують концепцію ізольованих браузерних профілів. Такі системи дозволяють створювати незалежні середовища роботи, керувати параметрами цифрової ідентифікації та запускати декілька профілів одночасно. Подібні рішення використовуються у веброзробці, тестуванні програмного забезпечення, дослідженні механізмів фінгерпринтингу та інших галузях, де необхідне розділення браузерних середовищ. [32–37]

Разом із перевагами ізоляція браузерних профілів має певні обмеження. По-перше, збільшення кількості профілів призводить до зростання обсягу локальних даних, які необхідно зберігати. По-друге, забезпечення узгодженості параметрів цифрової ідентифікації потребує додаткового контролю, оскільки

невідповідність між окремими характеристиками може сама по собі стати ознакою нестандартного середовища. Саме тому сучасні системи ізоляції часто поєднують механізми розділення даних із засобами керування цифровим відбитком браузера. [32–37]

Таким чином, ізоляція браузерних профілів є важливим інструментом забезпечення незалежності користувацьких середовищ у вебпросторі. Використання окремих сховищ даних, незалежних сесій та контрольованих параметрів цифрової ідентифікації дозволяє створювати відокремлені браузерні середовища, які можуть функціонувати незалежно одне від одного. Саме такі підходи використовуються в сучасних програмних засобах керування браузерними профілями та є основою для реалізації систем на кшталт IdentityLab.

1.4 Аналіз існуючих програмних засобів керування цифровою ідентифікацією

Зі зростанням складності механізмів вебідентифікації з’явилася потреба у програмних засобах, які дозволяють користувачам ізолювати браузерні середовища та керувати параметрами цифрової ідентифікації. Такі рішення отримали широке поширення серед веброзробників, тестувальників, фахівців з інформаційної безпеки та інших категорій користувачів, яким необхідно працювати з декількома незалежними браузерними профілями. [32–37]

Сучасні програмні засоби цього класу зазвичай поєднують механізми ізоляції локальних даних із можливістю зміни окремих параметрів браузерного середовища. Найчастіше користувачеві надається можливість керувати User-Agent, мовними налаштуваннями, часовим поясом, характеристиками графічної підсистеми та іншими параметрами, які можуть використовуватися для формування цифрового відбитка. [32–37]

Одним із найвідоміших представників даного класу програм є Multilogin. Основною особливістю цього рішення є підтримка великої кількості незалежних браузерних профілів та можливість централізованого керування ними. Кожен

профіль функціонує як окреме середовище з власними файлами cookie, локальними сховищами та налаштуваннями. Додатково користувач може змінювати параметри браузерного середовища та використовувати проксі-сервери для мережевої ізоляції. [32]

Іншим популярним рішенням є GoLogin. Даний програмний засіб орієнтований на створення ізольованих профілів з можливістю налаштування різноманітних характеристик браузера. Однією з його особливостей є наявність генератора профілів, який дозволяє автоматично формувати набір параметрів відповідно до обраної конфігурації. Також підтримується синхронізація профілів між різними пристроями користувача. [34]

Серед сучасних інструментів окреме місце займає AdsPower. Програмний засіб надає можливість створення великої кількості браузерних профілів та підтримує їх групове керування. Значна увага приділяється автоматизації роботи та інтеграції із зовнішніми сервісами. Крім того, користувач отримує засоби налаштування параметрів браузера та механізми розділення локальних даних між профілями. [33]

Ще одним відомим рішенням є Dolphin Anty. Основною сферою його використання є керування великою кількістю браузерних профілів та забезпечення їх ізоляції. Програмний засіб дозволяє створювати окремі середовища виконання, змінювати характеристики браузера та працювати з кількома профілями одночасно. Значна увага приділяється зручності керування профілями та швидкому перемиканню між ними. [35]

Окрім перелічених програмних продуктів, існують також інші рішення, зокрема Incogniton, Kameleo та ряд менш поширених систем. Незважаючи на відмінності в реалізації окремих функцій, більшість сучасних програмних засобів використовує схожі принципи роботи. Вони базуються на ізоляції браузерних профілів, незалежному зберіганні даних та керуванні параметрами цифрової ідентифікації. [36, 37]

Для більш наочного порівняння основних можливостей розглянутих програмних засобів доцільно сформувати узагальнену таблицю. [32–37]

Таблиця 1.1 – Порівняння функціональних можливостей програмних засобів

Функція	**Multilogin**	**GoLogin**	**AdsPower**	**Dolphin Anty**	**IdentityLab**
Ізольовані профілі	+	+	+	+	+
User-Agent Spoofing	+	+	+	+	+
Navigator Spoofing	+	+	+	+	+
Timezone Spoofing	+	+	+	+	+
WebGL Spoofing	+	+	+	+	+
Генерація профілів	+	+	+	+	+
Локальне зберігання профілів	+	+	+	+	+
Багатовкладковий режим	+	+	+	+	+

Наведене порівняння показує, що більшість сучасних програмних засобів використовує схожі підходи до ізоляції браузерних середовищ та керування параметрами цифрової ідентифікації. [32–37] Основні відмінності полягають у способах реалізації окремих функцій, зручності інтерфейсу, підтримці додаткових сервісів та особливостях архітектури.

Проведений аналіз дозволяє зробити висновок, що існуючі рішення забезпечують широкий набір інструментів для роботи з браузерними профілями, однак багато з них орієнтовані на комерційне використання та містять значну кількість функцій, які не завжди необхідні під час дослідження механізмів цифрової ідентифікації. Це обумовлює доцільність розробки власного програмного засобу, який дозволяє досліджувати принципи ізоляції браузерних профілів та керування параметрами цифрової ідентифікації на основі відкритих технологій Electron та Chromium. [19, 25]

Таким чином, аналіз сучасних програмних засобів показав, що ізоляція браузерних профілів та модифікація параметрів цифрової ідентифікації є

поширеними підходами до організації незалежних браузерних середовищ. Отримані результати стали основою для формування вимог до програмного засобу IdentityLab, який розглядається в наступних розділах роботи.

Висновок до першого розділу

У першому розділі було розглянуто теоретичні основи цифрової ідентифікації користувачів у веб-середовищі та проаналізовано сучасні підходи до формування браузерного відбитка. Встановлено, що сучасні вебресурси мають доступ до значної кількості технічних характеристик браузера та пристрою користувача, які можуть використовуватися для його розпізнавання та подальшого відстеження.

Дослідження показало, що формування цифрового відбитка здійснюється на основі сукупності різних параметрів, серед яких особливе значення мають User-Agent, характеристики об’єкта Navigator, параметри екрана, мовні налаштування, часовий пояс та дані графічної підсистеми, отримані за допомогою технології WebGL. Окремо було відзначено, що висока точність сучасних механізмів фінгерпринтингу досягається не за рахунок одного параметра, а шляхом аналізу великої кількості доступних характеристик браузерного середовища.

Також було розглянуто основні методи ізоляції браузерних профілів. Визначено, що найбільш ефективними підходами є використання незалежних сховищ даних, окремих браузерних сесій та відокремлених наборів параметрів цифрової ідентифікації. Такі механізми дозволяють забезпечити незалежність браузерних середовищ та мінімізувати взаємний вплив між різними профілями користувача.

У результаті аналізу сучасних програмних засобів було встановлено, що існуючі рішення реалізують схожі принципи роботи, які базуються на ізоляції профілів та керуванні параметрами цифрової ідентифікації. Водночас більшість таких систем є комерційними продуктами та орієнтовані на широкий спектр завдань, що створює передумови для розробки власного програмного засобу,

адаптованого до дослідження механізмів цифрової ідентифікації та ізоляції браузерних профілів.

Отримані результати стали основою для формування вимог до програмного засобу керування параметрами цифрової ідентифікації та ізоляції браузерних профілів IdentityLab. У наступному розділі буде розглянуто процес проєктування системи, її архітектуру, структуру браузерного профілю та механізми реалізації ізоляції й модифікації параметрів цифрової ідентифікації.

РОЗДІЛ 2
ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ IDENTITYLAB

2.1 Постановка задачі та вимоги до системи

Сучасні вебтехнології активно використовують механізми цифрової ідентифікації для персоналізації сервісів, забезпечення безпеки та аналізу поведінки користувачів. Під час відвідування вебресурсів браузер передає значну кількість параметрів, які можуть використовуватися для формування цифрового відбитка. До таких параметрів належать характеристики браузера, операційної системи, графічної підсистеми, мовних налаштувань, часової зони та інших компонентів програмного середовища. [1–5]

У процесі дослідження механізмів цифрової ідентифікації виникає необхідність створення контрольованого браузерного середовища, в якому можна змінювати окремі характеристики профілю та аналізувати їх вплив на формування браузерного відбитка. Використання звичайних браузерів не завжди дозволяє ефективно вирішувати такі завдання, оскільки вони мають обмежені можливості щодо ізоляції профілів та керування параметрами цифрової ідентифікації. [32–37]

Додатковою проблемою є необхідність одночасної роботи з кількома незалежними профілями. У разі використання спільного сховища даних різні профілі можуть впливати один на одного через файли cookie, локальні сховища або інші механізми збереження інформації. Це ускладнює проведення експериментів та знижує достовірність отриманих результатів.

Для вирішення зазначених проблем було прийнято рішення про розробку програмного засобу керування параметрами цифрової ідентифікації та ізоляції браузерних профілів IdentityLab. Програмний засіб призначений для створення незалежних браузерних профілів, керування їх характеристиками та дослідження механізмів формування цифрового відбитка у веб-середовищі.

Основною метою розробки є створення програмного засобу, який забезпечує ізольоване зберігання даних браузерних профілів та надає можливість

керування параметрами, що використовуються під час формування цифрової ідентифікації користувача.

Для досягнення поставленої мети було визначено такі функціональні вимоги до системи:

- створення браузерних профілів;
- редагування існуючих профілів;
- збереження профілів у файловій системі;
- запуск профілів у незалежних браузерних сесіях;
- підтримка багатовкладкового режиму роботи;
- генерація нових профілів на основі допустимих конфігурацій;
- налаштування параметрів User-Agent;
- налаштування параметрів Navigator API;
- налаштування мовних параметрів;
- налаштування часової зони;
- налаштування параметрів WebGL;
- аналіз параметрів браузерного відбитка;
- оцінювання рівня спостережуваності браузерного профілю за допомогою показника Risk Score.

Окрім функціональних вимог, до системи були сформульовані нефункціональні вимоги.

До основних нефункціональних вимог належать:

- забезпечення незалежності браузерних профілів;
- збереження працездатності вебресурсів після модифікації параметрів профілю;
- підтримка сучасних вебстандартів;
- можливість запуску на операційних системах сімейства Windows;
- зручність користувацького інтерфейсу;
- простота розширення функціональності програмного засобу;
- використання відкритих технологій та інструментів розробки.

Для реалізації поставлених вимог було обрано платформу Electron, яка поєднує можливості вебтехнологій із доступом до функціональності операційної системи. [19, 25, 28] Використання рушія Chromium дозволяє забезпечити сумісність із сучасними вебресурсами та використовувати механізми ізоляції браузерних сесій. У свою чергу середовище Node.js забезпечує роботу з файловою системою та реалізацію службової логіки застосунку.

У результаті аналізу предметної області та сформульованих вимог було визначено основні компоненти програмного засобу IdentityLab: менеджер браузерних профілів, модуль генерації профілів, браузерне середовище Browser Shell, модуль аналізу браузерного відбитка та підсистема оцінювання Risk Score. Взаємодія зазначених компонентів дозволяє створювати ізольовані браузерні профілі та досліджувати вплив різних параметрів на формування цифрової ідентифікації користувача.

Таким чином, сформульовані вимоги стали основою для подальшого проєктування архітектури програмного засобу IdentityLab та визначення структури його основних компонентів.

2.2 Архітектура програмного засобу

Під час проєктування програмного засобу IdentityLab основна увага приділялася забезпеченню ізоляції браузерних профілів, можливості керування параметрами цифрової ідентифікації та зручності подальшого розширення функціональності системи. Для реалізації поставлених вимог було обрано платформу Electron, яка дозволяє створювати настільні застосунки за допомогою вебтехнологій та використовувати рушій Chromium як основу браузерного середовища. [19, 25]

Архітектура IdentityLab побудована за модульним принципом. Кожен функціональний компонент відповідає за окрему частину логіки системи та взаємодіє з іншими компонентами через визначені механізми обміну даними. Такий підхід дозволяє спростити супровід програмного забезпечення та забезпечити можливість подальшого розвитку проєкту.

Основу системи становить головний процес Electron (Main Process), який відповідає за створення вікон застосунку, керування браузерними сесіями, взаємодію з файловою системою та виконання службових операцій. [19, 23] Саме в головному процесі здійснюється створення ізольованих браузерних профілів та керування їх життєвим циклом.

Користувацький інтерфейс реалізовано за допомогою HTML, CSS та JavaScript. [14, 29] Для побудови інтерфейсу не використовувалися додаткові фронтенд-фреймворки, що дозволило зменшити складність проєкту та зосередитися на реалізації основної функціональності. Інтерфейс забезпечує роботу з браузерними профілями, запуск браузерних сесій, аналіз параметрів цифрової ідентифікації та перегляд результатів тестування.

Одним із ключових компонентів системи є менеджер браузерних профілів. Його призначення полягає у створенні, редагуванні, видаленні та збереженні профілів користувача. Кожен профіль містить набір параметрів, які використовуються для формування цифрової ідентифікації браузера. Дані профілів зберігаються у вигляді JSON-файлів у файловій системі, що забезпечує простоту доступу до конфігурацій та можливість їх подальшого редагування. [30]

Для автоматизації процесу створення профілів у системі реалізовано окремий модуль генерації профілів. На відміну від підходу, що використовує фіксований набір готових конфігурацій, генератор формує профілі шляхом випадкового комбінування параметрів із наборів допустимих значень. При цьому враховуються правила сумісності між операційними системами, браузерами, мовними налаштуваннями та характеристиками обладнання.

Рисунок 2.1 – Схема роботи генератора браузерних профілів

Наприклад, під час генерації профілю для операційної системи macOS автоматично формуються параметри, характерні для пристроїв Apple, зокрема значення платформи MacIntel, відповідні мовні налаштування та сумісні конфігурації графічної підсистеми. Аналогічно для Windows використовуються інші набори параметрів, що відповідають типовим конфігураціям цієї операційної системи.

Для роботи з вебресурсами в IdentityLab реалізовано власне браузерне середовище Browser Shell. [22] Воно забезпечує відкриття вебсторінок, підтримку вкладок та взаємодію з браузерними профілями. Кожна вкладка працює в межах активного профілю та використовує його налаштування цифрової ідентифікації. Завдяки цьому користувач може одночасно працювати з декількома незалежними профілями без змішування даних між ними.

Окремим компонентом системи є модуль аналізу браузерного відбитка. Його завдання полягає у зборі інформації про поточні параметри браузерного середовища та відображенні їх користувачеві. Модуль дозволяє перевірити значення User-Agent, параметрів Navigator API, характеристик екрана, часової зони, графічної підсистеми та інших доступних вебресурсам даних.

Для оцінювання рівня спостережуваності браузерного профілю реалізовано підсистему Risk Score. Вона аналізує доступність основних fingerprint-сигналів та формує інтегральний показник, який відображає кількість параметрів, що можуть бути використані вебресурсом для формування цифрового відбитка. Важливо зазначити, що даний показник не оцінює якість профілю або ефективність його налаштувань, а характеризує лише рівень доступності ідентифікаційних сигналів для зовнішніх вебресурсів.

Загалом архітектура IdentityLab забезпечує поєднання механізмів ізоляції браузерних профілів, керування параметрами цифрової ідентифікації та аналізу браузерного відбитка в межах єдиного програмного середовища. Обрана структура системи дозволяє реалізувати необхідну функціональність та створює основу для подальшого розвитку програмного засобу.

2.3 Проєктування браузерного профілю

Одним із центральних елементів програмного засобу IdentityLab є браузерний профіль. Саме профіль визначає набір параметрів цифрової ідентифікації, які використовуються під час взаємодії браузера з вебресурсами. Від коректності формування профілю залежить узгодженість характеристик браузерного середовища та можливість створення ізольованих конфігурацій для проведення досліджень.

Під час проєктування системи було прийнято рішення зберігати профілі у вигляді JSON-документів. [30] Такий підхід забезпечує простоту структури даних, зручність читання та редагування конфігурацій, а також дозволяє легко додавати нові параметри без зміни загальної архітектури програмного засобу.

Кожний профіль має власний унікальний ідентифікатор та назву, що використовуються для його ідентифікації в системі. Ідентифікатор дозволяє однозначно пов’язати профіль із відповідною браузерною сесією, а назва використовується для відображення в інтерфейсі застосунку.

Структура профілю складається з декількох логічних блоків, кожен з яких відповідає за окрему групу параметрів цифрової ідентифікації.

Перший блок містить характеристики середовища Navigator. Саме ці параметри найчастіше використовуються вебресурсами для визначення операційної системи, браузера та апаратних характеристик пристрою. До складу блоку входять значення User-Agent, основна мова браузера, список доступних мов, тип платформи, кількість логічних процесорів, обсяг оперативної пам’яті та інші параметри, доступні через Navigator API. [7]

Другий блок містить характеристики екрана. До нього входять роздільна здатність дисплея, доступна область екрана, глибина кольору та значення devicePixelRatio. Ці параметри використовуються вебресурсами для адаптації інтерфейсу та одночасно можуть брати участь у формуванні браузерного відбитка. [11]

Окремий блок профілю відповідає за часові та регіональні налаштування. У ньому зберігається інформація про часовий пояс та його зміщення відносно UTC. Узгодженість часового поясу з мовними параметрами та операційною системою є важливою умовою формування логічно коректного профілю.

Наступним елементом структури є блок параметрів локального зберігання даних. Він містить інформацію про доступність LocalStorage, SessionStorage, IndexedDB та інших механізмів збереження даних, що використовуються сучасними вебдодатками. Наявність цих компонентів дозволяє браузеру коректно працювати з більшістю сучасних вебресурсів. [12, 13, 15]

Важливу роль у структурі профілю відіграє графічний блок. Він містить параметри Canvas та WebGL, зокрема інформацію про виробника графічного адаптера та використовуваний механізм рендерингу. Оскільки характеристики графічної підсистеми мають високу інформативність під час побудови браузерного відбитка, даний блок є одним із ключових елементів профілю. [8, 9, 17]

Також у структурі профілю передбачено блок мультимедійних можливостей браузера. До нього входять відомості про підтримку AudioContext та WebRTC. Дані компоненти широко використовуються сучасними

вебдодатками і можуть виступати додатковими джерелами інформації для аналізу цифрової ідентифікації. [10, 16]

Окремо зберігається інформація про дозволи браузера. До таких дозволів належать доступ до геолокації, сповіщень, камери та мікрофона. Наявність цих параметрів дозволяє аналізувати поведінку браузерного середовища під час взаємодії з вебресурсами.

Для спрощення створення нових профілів у системі реалізовано механізм автоматичної генерації конфігурацій. На відміну від використання фіксованого набору готових шаблонів, генератор формує профілі шляхом випадкового вибору параметрів із наборів допустимих значень. При цьому застосовуються правила сумісності між окремими характеристиками.

Наприклад, під час формування профілю для операційної системи macOS автоматично використовуються значення платформи MacIntel, відповідні мовні налаштування та сумісні конфігурації графічної підсистеми. Аналогічно для Windows формуються параметри, характерні для середовища Windows та браузерів, що використовуються на цій платформі. Такий підхід дозволяє отримувати різноманітні профілі без виникнення очевидних суперечностей між окремими характеристиками.

У результаті проведеного проєктування було сформовано структуру браузерного профілю, яка забезпечує зберігання параметрів цифрової ідентифікації, підтримує механізми ізоляції браузерних середовищ та дозволяє реалізувати подальші засоби аналізу і модифікації цифрового відбитка в програмному засобі IdentityLab.

2.4 Реалізація механізмів ізоляції браузерних профілів

Однією з основних вимог до програмного засобу IdentityLab була можливість одночасної роботи з кількома незалежними браузерними профілями без змішування їхніх локальних даних. Для досягнення цієї мети було реалізовано механізм ізоляції профілів на основі сесій Chromium, доступних через платформу Electron. [19, 20]

У стандартному режимі роботи браузер використовує єдине сховище даних, у якому зберігаються файли cookie, кеш, локальні бази даних та інша службова інформація. Такий підхід є зручним для повсякденного використання, однак не дозволяє створювати повністю незалежні середовища виконання в межах одного застосунку. Саме тому під час розробки IdentityLab було прийнято рішення використовувати окремі сесії для кожного браузерного профілю.

Платформа Electron надає механізм Session API, який дозволяє створювати незалежні браузерні сесії за допомогою ідентифікаторів partition. [20] Кожна така сесія має власний набір локальних даних і функціонує незалежно від інших сесій застосунку.

Для кожного профілю IdentityLab формується унікальний ідентифікатор сесії у форматі:

persist:profile-<profileId>,

де profileId являє собою унікальний ідентифікатор браузерного профілю.

Під час запуску профілю застосунок створює окрему браузерну сесію за допомогою механізму session.fromPartition(). [20] Використання префікса persist забезпечує збереження даних між перезапусками застосунку та дозволяє профілю використовувати власне постійне сховище.

Після створення сесії інформація про її ідентифікатор передається до браузерного середовища Browser Shell. Для цього використовується механізм параметрів запуску вікна браузера, що дозволяє передати активний partition до інтерфейсу браузера.

Під час створення нової вкладки Browser Shell використовує отриманий ідентифікатор сесії та призначає його відповідному компоненту webview. [22] У результаті всі вкладки, відкриті в межах конкретного профілю, працюють через одну й ту саму ізольовану браузерну сесію та мають доступ лише до даних власного профілю.

Використання окремих partition дозволяє забезпечити незалежне зберігання таких даних:

- файлів cookie;

- LocalStorage;
- SessionStorage;
- IndexedDB;
- кешу браузера;
- службових даних Chromium-сесії.

Завдяки цьому авторизація в одному профілі не впливає на інші профілі навіть у випадку роботи з одним і тим самим вебресурсом. Наприклад, користувач може одночасно відкрити декілька профілів та виконати вхід до різних облікових записів одного сервісу без конфліктів між сесіями.

Важливою перевагою обраного підходу є його інтеграція з механізмами Chromium. Оскільки ізоляція реалізується на рівні браузерного рушія, вебресурси взаємодіють із профілями як із незалежними браузерними середовищами. [25, 26] Це забезпечує високу сумісність із сучасними вебдодатками та не потребує додаткового втручання в логіку роботи сайтів.

Ще однією перевагою є можливість збереження стану профілю між запусками програми. Завдяки використанню persistent-сесій користувач не втрачає авторизацію, локальні налаштування та інші дані після закриття застосунку. Це робить роботу з профілями більш зручною та наближеною до поведінки звичайного браузера.

Таким чином, механізм ізоляції браузерних профілів у програмному засобі IdentityLab реалізовано на основі Session API платформи Electron та окремих persistent partition для кожного профілю. Такий підхід забезпечує незалежне зберігання даних, підтримує одночасну роботу декількох профілів і створює основу для подальшого керування параметрами цифрової ідентифікації.

2.5 Реалізація механізмів керування параметрами цифрової ідентифікації

Після реалізації механізмів ізоляції браузерних профілів наступним етапом розробки програмного засобу IdentityLab стало впровадження засобів керування параметрами цифрової ідентифікації. Основною метою даного механізму є можливість формування контрольованого браузерного середовища,

характеристики якого можуть відрізнятися від реальних параметрів комп’ютера користувача.

Як було встановлено в першому розділі, сучасні вебресурси використовують значну кількість характеристик браузера для формування цифрового відбитка. [1–5] Тому для проведення досліджень необхідно забезпечити можливість зміни найбільш важливих параметрів, які впливають на процес цифрової ідентифікації.

У програмному засобі IdentityLab реалізовано механізми керування User-Agent, параметрами Navigator API, часовою зоною та характеристиками графічної підсистеми WebGL. Зазначені компоненти були обрані через їх значний вплив на формування браузерного відбитка та широке використання сучасними вебресурсами.

User-Agent Spoofing

Одним із базових елементів цифрової ідентифікації є рядок User-Agent. Він використовується вебресурсами для визначення браузера, його версії та операційної системи користувача. [14]

У IdentityLab для кожного профілю може бути визначене власне значення User-Agent, яке застосовується під час відкриття вебсторінок. Завдяки цьому браузерне середовище може імітувати роботу різних конфігурацій браузерів та операційних систем.

Під час генерації профілів значення User-Agent формується відповідно до обраної операційної системи та браузера. Такий підхід дозволяє уникнути очевидних невідповідностей між різними параметрами профілю та забезпечити логічну узгодженість конфігурації.

Navigator Spoofing

Значна частина інформації про браузерне середовище доступна через об’єкт Navigator. Саме тому в IdentityLab реалізовано можливість керування низкою його властивостей.

До параметрів, які можуть змінюватися відповідно до конфігурації профілю, належать:

- navigator.platform;
- navigator.language;
- navigator.languages;
- navigator.hardwareConcurrency;
- navigator.deviceMemory;
- navigator.maxTouchPoints;
- window.devicePixelRatio.

Зазначені параметри використовуються для відображення характеристик операційної системи, апаратної конфігурації пристрою та мовного середовища. [7] Під час створення профілю вони підбираються відповідно до правил сумісності між платформою, регіоном та характеристиками обладнання.

Наприклад, для профілю, що імітує роботу macOS, використовуються значення платформи MacIntel, відповідні мовні налаштування та характерні параметри дисплея. Аналогічно для Windows застосовуються конфігурації, характерні для даної операційної системи.

Timezone Spoofing

Часовий пояс також є важливою складовою цифрової ідентифікації користувача. Багато вебресурсів враховують його під час аналізу браузерного середовища та перевірки узгодженості параметрів профілю. [6, 7]

У програмному засобі реалізовано можливість встановлення довільного значення часової зони для кожного профілю. Під час генерації конфігурацій часовий пояс підбирається відповідно до обраного регіону та мовних налаштувань.

Наприклад, для профілів, орієнтованих на Україну, використовується часова зона Europe/Kyiv та відповідні мовні параметри. Це дозволяє забезпечити узгодженість між географічними та мовними характеристиками профілю.

WebGL Spoofing

Одним із найбільш інформативних джерел даних для формування цифрового відбитка є технологія WebGL. Через спеціальні програмні інтерфейси

вебресурси можуть отримувати відомості про графічну підсистему пристрою, включаючи виробника графічного адаптера та механізм рендерингу.

У програмному засобі IdentityLab реалізовано механізм керування такими параметрами:

- WebGL Vendor;
- WebGL Renderer. [9, 17, 18]

Зміна цих характеристик дозволяє формувати різні конфігурації графічного середовища відповідно до обраного профілю. Під час генерації конфігурацій використовуються значення, характерні для реальних апаратних платформ, що додатково підвищує узгодженість браузерного профілю.

Узгодженість параметрів профілю

Важливим аспектом реалізації механізмів керування цифровою ідентифікацією є забезпечення логічної узгодженості між усіма характеристиками профілю. Під час проєктування системи було прийнято рішення не використовувати випадкову зміну окремих параметрів без урахування їх взаємозв’язків.

Замість цього генератор профілів використовує набір правил сумісності між операційними системами, браузерами, мовними параметрами, часовими зонами та характеристиками обладнання. Завдяки цьому сформовані профілі відповідають реальним конфігураціям пристроїв та не містять очевидних суперечностей.

Наприклад, профіль macOS не може отримати значення платформи Win32 або часовий пояс, який не відповідає обраному регіону. Аналогічно конфігурації графічної підсистеми підбираються відповідно до характеристик цільової платформи.

Таким чином, реалізовані в IdentityLab механізми керування параметрами цифрової ідентифікації забезпечують можливість формування контрольованих браузерних профілів із узгодженими характеристиками. Поєднання User-Agent Spoofing, Navigator Spoofing, Timezone Spoofing та WebGL Spoofing дозволяє створювати різноманітні конфігурації браузерного середовища та

використовувати їх для дослідження механізмів цифрової ідентифікації у вебпросторі.

Висновок до другого розділу

У другому розділі було виконано проєктування програмного засобу керування параметрами цифрової ідентифікації та ізоляції браузерних профілів IdentityLab. На основі сформульованих вимог визначено основні компоненти системи, їх призначення та принципи взаємодії.

У результаті проєктування було обрано платформу Electron, яка використовує рушій Chromium та середовище Node.js. Такий підхід дозволив поєднати можливості сучасного браузера з функціями настільного застосунку та забезпечити доступ до механізмів керування браузерними сесіями.

Було розроблено структуру браузерного профілю, яка забезпечує зберігання параметрів цифрової ідентифікації у форматі JSON. До складу профілю включено характеристики браузера, операційної системи, екрана, часової зони, графічної підсистеми та інших компонентів, що можуть використовуватися для формування браузерного відбитка.

Окрему увагу приділено реалізації механізмів ізоляції браузерних профілів. Для забезпечення незалежності середовищ використано Session API платформи Electron та механізм persistent partition. Такий підхід дозволяє ізолювати файли cookie, локальні сховища даних, кеш та іншу інформацію, що накопичується під час роботи браузера.

Також було реалізовано механізми керування параметрами цифрової ідентифікації, які включають зміну User-Agent, параметрів Navigator API, часової зони та характеристик WebGL. Для формування конфігурацій використовується генератор профілів, який забезпечує узгодженість між окремими параметрами та дозволяє створювати реалістичні браузерні середовища.

Отримані результати проєктування стали основою для реалізації програмного засобу IdentityLab. У наступному розділі розглядаються особливості практичної реалізації системи, її основних модулів та результати експериментального дослідження.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ ПРОГРАМНОГО ЗАСОБУ

3.1 Реалізація менеджера браузерних профілів.

Одним із ключових компонентів програмного засобу IdentityLab є менеджер браузерних профілів, який забезпечує централізоване керування створеними конфігураціями. Саме через даний модуль користувач взаємодіє з профілями, створює нові конфігурації, запускає браузерні сесії та виконує аналіз параметрів цифрової ідентифікації.

Головне вікно застосунку реалізовано у вигляді єдиного робочого простору, у якому відображаються наявні профілі та доступні дії над ними. Для кожного профілю відображаються основні характеристики, зокрема назва профілю, операційна система, браузер, країна та поточна оцінка Risk Score за наявності збереженого результату аналізу.

На головному екрані користувачеві доступний набір основних операцій:

- створення нового профілю;
- автоматична генерація профілю;
- відкриття профілю;
- очищення локальних даних профілю;
- видалення профілю;
- запуск аналізу браузерного відбитка.

Створення нового профілю виконується через окрему форму налаштування параметрів. Користувач може самостійно визначити основні характеристики майбутнього профілю, включаючи операційну систему, браузер, мовні параметри, часову зону та характеристики графічного середовища. Після завершення налаштування конфігурація зберігається у файловій системі та стає доступною для подальшого використання.

Для спрощення роботи реалізовано механізм автоматичної генерації профілів. На відміну від використання готових шаблонів, генератор формує конфігурацію шляхом випадкового вибору параметрів із наборів допустимих значень з урахуванням правил сумісності. Це дозволяє швидко створювати

різноманітні профілі без необхідності ручного налаштування кожного параметра.

Після створення профілю користувач може запустити його у браузерному середовищі Browser Shell. У цьому випадку система створює окрему ізольовану сесію Chromium та відкриває нове вікно браузера, яке використовує параметри обраного профілю.

Окремою функцією менеджера є очищення даних профілю. Дана операція дозволяє видалити накопичені локальні дані, включаючи файли cookie, кеш браузера та інші дані сесії. Це може бути корисним під час повторного тестування профілю або проведення експериментів із чистим браузерним середовищем.

Для повного видалення конфігурації передбачено функцію видалення профілю. У результаті її виконання з файлової системи вилучаються дані профілю та пов’язані з ним налаштування.

Ще однією важливою можливістю менеджера профілів є запуск модуля Fingerprint Analyzer. На відміну від Browser Shell, який призначений для взаємодії з вебресурсами, модуль аналізу використовується для збору та дослідження параметрів цифрової ідентифікації поточного профілю. Запуск аналізатора здійснюється безпосередньо з головного вікна застосунку та відкриває окреме вікно для перегляду результатів.

Таким чином, реалізований менеджер профілів забезпечує централізоване керування конфігураціями браузерних середовищ та об’єднує всі основні функції програмного засобу IdentityLab. Його використання дозволяє спростити роботу з профілями та забезпечує швидкий доступ до інструментів ізоляції та аналізу цифрової ідентифікації.

Рисунок 3.1 - Головне вікно програмного засобу IdentityLab

3.2 Реалізація Browser Shell

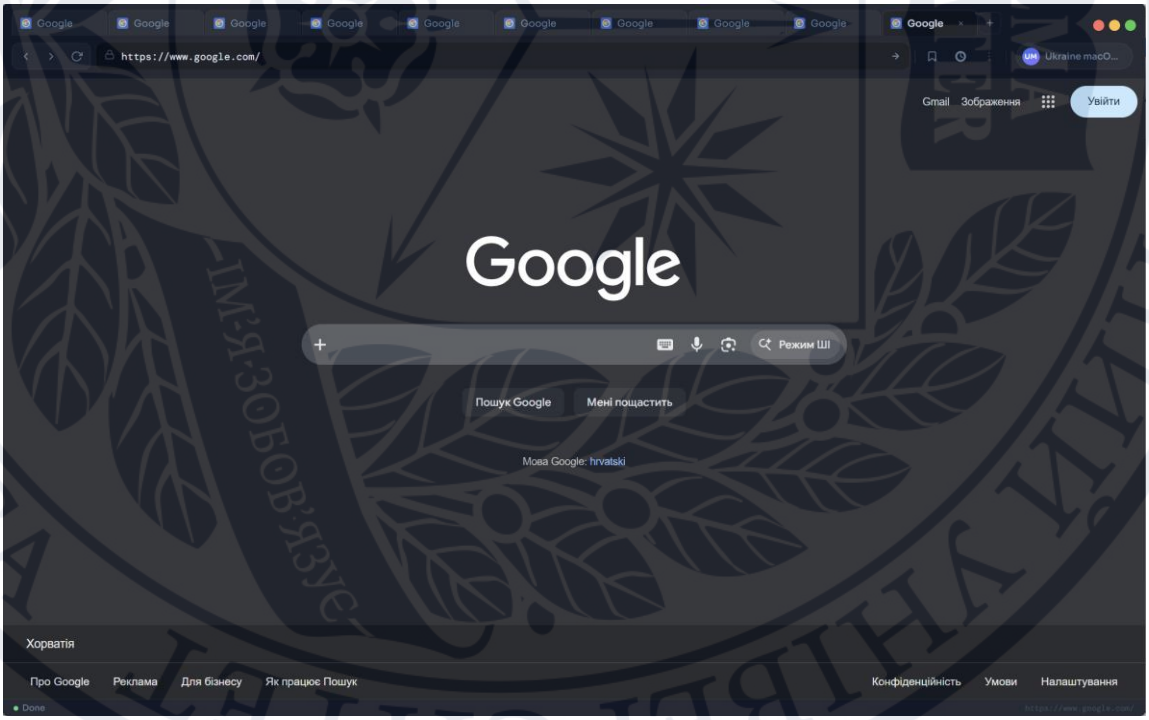

Рисунок 3.2 - Вікно Browser Shell програмного засобу IdentityLab

Для взаємодії з веб-ресурсами в програмному засобі IdentityLab реалізовано власне браузерне середовище Browser Shell. Його основним

призначенням є відкриття веб-сторінок із використанням параметрів обраного браузерного профілю та забезпечення роботи користувача в межах ізольованої Chromium-сесії.

Browser Shell запускається безпосередньо з менеджера профілів після вибору команди відкриття профілю. Під час запуску система створює окреме вікно браузера та передає йому параметри активного профілю, включаючи ідентифікатор ізольованої сесії. Завдяки цьому всі вкладки, відкриті в межах вікна браузера, використовують налаштування конкретного профілю та мають доступ лише до його локальних даних.

Інтерфейс Browser Shell побудовано за принципами класичних сучасних браузерів. У верхній частині вікна розташовано панель вкладок, яка дозволяє працювати з кількома вебсторінками одночасно. Кожна вкладка містить власний екземпляр компонента webview та може незалежно завантажувати вебресурси.

Для реалізації багатовкладкового режиму передбачено механізм створення нових вкладок. Натискання кнопки додавання вкладки створює новий екземпляр webview у межах активної браузерної сесії. Закриття вкладки здійснюється за допомогою окремої кнопки, розташованої безпосередньо на вкладці. Перемикання між вкладками виконується шляхом вибору потрібної вкладки на панелі навігації.

Під панеллю вкладок розміщено адресний рядок та елементи керування навігацією. До складу навігаційної панелі входять:

- кнопка переходу на попередню сторінку;
- кнопка переходу на наступну сторінку;
- кнопка оновлення сторінки;
- кнопка переходу на домашню сторінку;
- адресний рядок для введення URL-адреси.

Зазначені елементи забезпечують базову функціональність браузера та дозволяють користувачу здійснювати навігацію між веб-ресурсами без використання зовнішніх програм.

Особливістю Browser Shell є відображення активного браузерного профілю. Для цього в інтерфейсі використовується спеціальний елемент Profile Badge, який містить назву поточного профілю. Це дозволяє користувачеві швидко визначити, в якому саме середовищі виконується робота, що особливо важливо під час одночасного використання декількох профілів.

Відображення вебсторінок здійснюється за допомогою компонента webview, який є частиною платформи Electron. Даний компонент використовує можливості рушія Chromium та забезпечує сумісність із сучасними веб-технологіями. Кожен webview отримує ідентифікатор активної сесії та використовує її локальні дані, включаючи файли cookie, LocalStorage та інші елементи браузерного середовища.

Під час відкриття веб-сторінок Browser Shell автоматично застосовує параметри цифрової ідентифікації, визначені в конфігурації профілю. У результаті вебресурси отримують значення User-Agent, мовних налаштувань, часової зони та інших характеристик відповідно до активного профілю. Це дозволяє використовувати Browser Shell як середовище для дослідження впливу різних параметрів на формування браузерного відбитка.

Завдяки поєднанню ізольованих Chromium-сесій, багатовкладкового режиму та підтримки механізмів керування цифровою ідентифікацією Browser Shell виступає основним інструментом взаємодії користувача з вебпростором у межах програмного засобу IdentityLab.

3.3 Реалізація Fingerprint Analyzer

Рисунок 3.3 - Вікно Fingerprint Analyzer

Для дослідження параметрів цифрової ідентифікації в програмному засобі IdentityLab реалізовано окремий модуль Fingerprint Analyzer. Його основним призначенням є збір та відображення характеристик браузерного середовища, які можуть використовуватися вебресурсами під час формування браузерного відбитка.

На відміну від Browser Shell, який використовується для безпосередньої роботи з вебсторінками, Fingerprint Analyzer виконує роль інструмента дослідження та аналізу. Запуск модуля здійснюється з менеджера профілів за допомогою команди аналізу профілю. Після запуску відкривається окреме вікно, у якому відображаються результати збору інформації про поточне браузерне середовище.

Під час виконання аналізу система збирає набір параметрів, які найчастіше використовуються сучасними механізмами браузерного фінгерпринтингу. Отримані дані групуються за декількома категоріями для спрощення подальшого аналізу.

До першої категорії належать параметри Navigator API. У межах даного блоку збираються відомості про браузерне середовище, включаючи значення User-Agent, основну мову браузера, список доступних мов, платформу, кількість логічних процесорів, обсяг оперативної пам’яті та інші характеристики, доступні через об’єкт Navigator.

Наступний блок містить характеристики екрана. До нього входять роздільна здатність дисплея, доступна область екрана, глибина кольору, параметри браузерного вікна та значення devicePixelRatio. Дані характеристики можуть використовуватися вебресурсами для формування додаткових ознак браузерного відбитка.

Окремо збирається інформація про часову зону та її зміщення відносно координованого всесвітнього часу. Даний параметр дозволяє перевірити відповідність регіональних налаштувань конфігурації браузерного профілю.
Для аналізу локальних механізмів зберігання даних модуль перевіряє доступність LocalStorage, SessionStorage, IndexedDB та файлів cookie. Наявність цих компонентів є важливою характеристикою сучасного браузерного середовища та впливає на можливості роботи вебдодатків.

Особливу увагу приділено графічній підсистемі браузера. У межах аналізу визначається доступність WebGL, а також збирається інформація про виробника графічного адаптера та механізм рендерингу. Додатково формується Canvas Hash, який може використовуватися як одна з характеристик браузерного відбитка.

Також модуль аналізує підтримку мультимедійних компонентів браузера. До них належать AudioContext та WebRTC. Дані технології широко використовуються сучасними вебресурсами та можуть виступати додатковими джерелами інформації під час формування цифрової ідентифікації користувача.

Окрім технічних характеристик браузера, система виконує збір інформації про дозволи, доступні вебресурсам. До таких дозволів належать доступ до геолокації, сповіщень, камери та мікрофона. Отримані результати дозволяють

оцінити рівень взаємодії браузера з операційною системою та доступність окремих функцій для вебдодатків.

Після завершення збору даних усі результати формуються у структурований JSON-звіт. Використання формату JSON забезпечує простоту обробки інформації та дозволяє використовувати результати аналізу в інших компонентах системи. Звіт містить повний набір зібраних характеристик, а також службову інформацію про профіль і час виконання аналізу.

Отримані результати використовуються не лише для відображення користувачеві, а й для подальшого розрахунку показника Risk Score. Таким чином, Fingerprint Analyzer виступає центральним компонентом системи аналізу цифрової ідентифікації та забезпечує збір інформації, необхідної для оцінювання характеристик браузерного середовища.

3.4 Реалізація механізму Risk Score

Під час розробки програмного засобу IdentityLab було реалізовано механізм Risk Score, призначений для оцінювання рівня спостережуваності браузерного профілю. На відміну від класичних систем оцінювання ризиків інформаційної безпеки, даний показник не визначає рівень захищеності користувача та не використовується для виявлення загроз. Його основне призначення полягає в оцінюванні кількості параметрів браузерного середовища, які потенційно доступні вебресурсам під час формування браузерного відбитка.

Основою механізму є rule-based модель, що використовує набір правил для аналізу доступних fingerprint-сигналів. Кожна категорія параметрів має власну вагу, яка додається до загального результату за умови доступності відповідного сигналу для вебресурсу.

У поточній реалізації аналізуються такі категорії параметрів:

- доступність WebGL;
- доступність Canvas Fingerprinting;
- доступність інформації про кількість логічних процесорів;
- доступність інформації про обсяг оперативної пам’яті;

- доступність параметрів екрана;
- доступність часового поясу;
- доступність мовних налаштувань;
- підтримка WebRTC.

Для кожної категорії встановлено певну кількість балів. Під час виконання аналізу система перевіряє наявність відповідних даних та формує підсумкове значення Risk Score у діапазоні від 0 до 100 балів.

Одночасно з числовою оцінкою формується перелік причин, які вплинули на результат. Це дозволяє користувачеві не лише отримати підсумковий показник, а й зрозуміти, які саме характеристики браузерного середовища були враховані під час аналізу.

Прикладом таких причин можуть бути:

- WebGL доступний;
- Canvas Fingerprint доступний;
- доступна інформація про процесор;
- доступна інформація про пам’ять;
- доступні параметри екрана;
- доступний часовий пояс;
- доступний список мов;
- підтримується WebRTC.

Збереження переліку причин є важливою перевагою реалізованого підходу, оскільки дозволяє пояснити механізм формування оцінки та підвищує прозорість роботи системи.

Важливо зазначити, що високий показник Risk Score не свідчить про низьку ефективність програмного засобу або некоректну конфігурацію профілю. У межах IdentityLab цей показник використовується виключно для оцінювання кількості доступних fingerprint-сигналів, а не для визначення рівня анонімності користувача.

Під час проєктування системи було прийнято рішення не блокувати повністю механізми, які активно використовуються сучасними веб-додатками.

Повне вимкнення WebGL, Canvas, AudioContext або інших програмних інтерфейсів могло б негативно вплинути на працездатність вебресурсів та створити нетипову конфігурацію браузерного середовища. У деяких випадках така поведінка сама по собі може бути додатковою ознакою нестандартного браузера.

Замість приховування відповідних параметрів у програмному засобі використовується підхід, заснований на керуванні їх значеннями. Вебресурс продовжує отримувати доступ до необхідних характеристик браузера, однак замість реальних параметрів пристрою використовуються значення, визначені конфігурацією профілю.

Таким чином, високий показник Risk Score означає, що вебресурс має доступ до значної кількості параметрів браузерного середовища, проте це не означає використання реальних характеристик пристрою користувача. Основним завданням IdentityLab є створення контрольованого та узгодженого браузерного профілю для дослідження механізмів цифрової ідентифікації, а не повне приховування всіх доступних fingerprint-сигналів.

У результаті реалізації механізму Risk Score користувач отримує додатковий інструмент для аналізу браузерного середовища та оцінювання кількості параметрів, які можуть бути використані під час формування цифрового відбитка. Це дозволяє більш детально досліджувати вплив окремих характеристик на процес цифрової ідентифікації у веб-середовищі.

3.5 Експериментальне дослідження та тестування

Рисунок 3.4 - Результати аналізу профілю у Fingerprint Analyzer

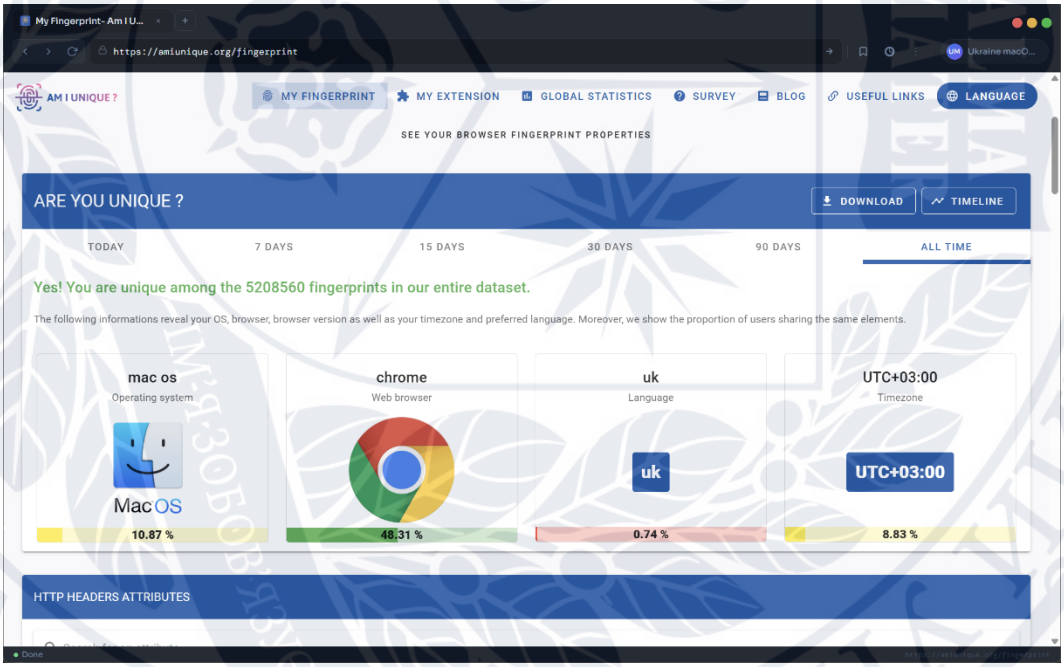

Рисунок 3.5 – Результати тестування профілю за допомогою сервісу AmIUnique

Після завершення реалізації програмного засобу IdentityLab було проведено експериментальне дослідження його функціональності. Основною метою тестування стала перевірка коректності роботи механізмів ізоляції

браузерних профілів, керування параметрами цифрової ідентифікації та аналізу браузерного відбитка.

Дослідження проводилося на основі профілів, сформованих за допомогою вбудованого генератора конфігурацій. Для кожного профілю виконувалася перевірка узгодженості параметрів браузерного середовища, а також аналіз доступних fingerprint-сигналів.

На першому етапі тестування використовувався модуль Fingerprint Analyzer, реалізований безпосередньо в IdentityLab. За допомогою даного інструмента здійснювався збір параметрів браузерного середовища та формування структурованого звіту у форматі JSON.

У процесі аналізу було підтверджено коректне застосування параметрів профілю, включаючи:

- значення User-Agent;
- мовні налаштування;
- параметри платформи;
- характеристики екрана;
- часовий пояс;
- параметри WebGL;
- підтримку локальних сховищ даних.

Отримані результати показали, що браузерне середовище використовує характеристики, визначені конфігурацією профілю, а не реальні параметри пристрою користувача.

Окремо було виконано перевірку механізму Risk Score. Під час аналізу система формувала числову оцінку та перелік причин, які вплинули на результат. Для досліджуваного профілю були виявлені такі доступні категорії fingerprint-сигналів:

- WebGL;
- Canvas Fingerprinting;
- інформація про процесор;
- інформація про пам’ять;

- параметри екрана;
- часовий пояс;
- мовні налаштування;
- WebRTC.

На основі отриманих даних було сформовано високий показник Risk Score. Водночас результати аналізу підтвердили, що високий рівень оцінки обумовлений доступністю відповідних сигналів для вебресурсів, а не використанням реальних характеристик пристрою. Це відповідає закладеній концепції IdentityLab, згідно з якою система керує параметрами цифрової ідентифікації замість повного блокування окремих механізмів браузера.

Другий етап дослідження передбачав використання зовнішніх сервісів аналізу браузерного відбитка. Для цього було використано сервіс AmIUnique, який дозволяє оцінити унікальність конфігурації браузера та проаналізувати характеристики, доступні для формування цифрового відбитка.

Під час тестування було підтверджено, що параметри браузерного середовища відповідають налаштуванням активного профілю. Зокрема, сервіс коректно визначав операційну систему, браузер, мовні налаштування та інші характеристики відповідно до конфігурації профілю, а не до реального середовища виконання застосунку.

Окрему увагу було приділено перевірці механізмів ізоляції браузерних профілів. Для цього виконувалося послідовне відкриття декількох профілів та перевірка незалежності локальних даних. У результаті тестування було встановлено, що файли cookie, LocalStorage, SessionStorage та інші елементи локального сховища не передаються між профілями та залишаються ізольованими в межах відповідних Chromium-сесій.

Проведене тестування також підтвердило працездатність Browser Shell під час роботи з декількома вкладками. Вкладки коректно використовували параметри активного профілю та працювали в межах відповідної ізольованої сесії без втрати функціональності вебресурсів.

Отримані результати дозволяють зробити висновок, що реалізовані в IdentityLab механізми ізоляції браузерних профілів та керування параметрами цифрової ідентифікації працюють відповідно до поставлених вимог. Система забезпечує незалежність браузерних середовищ, дозволяє формувати узгоджені конфігурації профілів та надає інструменти для аналізу характеристик цифрової ідентифікації у веб-середовищі.

Висновок до третього розділу

У третьому розділі було розглянуто особливості реалізації програмного засобу IdentityLab та проведено дослідження його функціональних можливостей. Реалізовано менеджер браузерних профілів, який забезпечує створення, генерацію, відкриття, очищення та видалення профілів, а також запуск інструментів аналізу цифрової ідентифікації.

Для роботи з вебресурсами розроблено браузерне середовище Browser Shell, яке підтримує багатовкладковий режим, навігацію між сторінками та використання ізольованих Chromium-сесій. Завдяки інтеграції з механізмами керування параметрами цифрової ідентифікації браузерне середовище автоматично застосовує налаштування активного профілю під час взаємодії з вебресурсами.

Окрему увагу приділено реалізації модуля Fingerprint Analyzer, який виконує збір та аналіз параметрів браузерного середовища. Модуль забезпечує формування структурованого звіту про характеристики браузера, операційної системи, графічної підсистеми, локальних сховищ даних та інших компонентів, що можуть використовуватися під час формування браузерного відбитка.

Також було реалізовано механізм Risk Score, призначений для оцінювання рівня доступності fingerprint-сигналів для вебресурсів. На відміну від традиційних оцінок ризику, даний показник не визначає рівень захищеності користувача, а характеризує кількість параметрів браузерного середовища, доступних для аналізу. Використання підходу, заснованого на керуванні значеннями параметрів, дозволяє зберігати сумісність із сучасними

вебресурсами без повного блокування окремих програмних інтерфейсів браузера.

У результаті експериментального дослідження підтверджено коректність роботи механізмів ізоляції браузерних профілів та модифікації параметрів цифрової ідентифікації. Проведене тестування показало, що створені профілі використовують задані характеристики браузерного середовища, а локальні дані залишаються ізольованими в межах відповідних Chromium-сесій.

Отримані результати свідчать про досягнення поставленої мети розробки та підтверджують можливість використання програмного засобу IdentityLab для дослідження механізмів цифрової ідентифікації та ізоляції браузерних профілів у веб-середовищі.

ВИСНОВКИ

У бакалаврській роботі розглянуто проблему цифрової ідентифікації користувачів у веб-середовищі та досліджено сучасні методи формування браузерного відбитка. Проведений аналіз показав, що сучасні вебресурси можуть використовувати значну кількість параметрів браузера та пристрою для ідентифікації користувачів, зокрема характеристики браузера, операційної системи, графічної підсистеми, мовних налаштувань та інших компонентів веб-середовища.

У процесі виконання роботи було проаналізовано принципи браузерного фінгерпринтингу, особливості використання браузерних профілів та можливості сучасних засобів керування цифровою ідентифікацією. Також проведено огляд існуючих рішень для роботи з браузерними профілями та визначено їх основні переваги й обмеження.

На основі проведеного аналізу було сформульовано вимоги до програмного засобу для ізоляції браузерних профілів та керування параметрами цифрової ідентифікації. У результаті виконано проєктування архітектури системи IdentityLab, визначено структуру браузерного профілю та реалізовано механізми взаємодії між основними компонентами програмного забезпечення.

У ході практичної частини роботи розроблено програмний засіб IdentityLab на базі платформи Electron та рушія Chromium. Реалізовано менеджер браузерних профілів, який забезпечує створення, генерацію, відкриття, очищення та видалення профілів. Для зберігання конфігурацій використано файлову систему та формат JSON, що забезпечує простоту розширення та супроводу системи.

Одним із ключових результатів роботи стала реалізація механізму ізоляції браузерних профілів на основі Session API платформи Electron та окремих persistent-сесій Chromium. Використаний підхід дозволяє забезпечити незалежне зберігання локальних даних профілів, включаючи файли cookie, локальні сховища даних та інші елементи браузерного середовища.

Також реалізовано механізми керування параметрами цифрової ідентифікації, зокрема User-Agent Spoofing, Navigator Spoofing, Timezone Spoofing та WebGL Spoofing. Для формування профілів створено rule-based генератор, який враховує сумісність між операційними системами, браузерами, мовними налаштуваннями та характеристиками обладнання.

У складі програмного засобу розроблено модуль Fingerprint Analyzer, призначений для збору та аналізу параметрів браузерного середовища. Додатково реалізовано механізм Risk Score, який дозволяє оцінювати кількість fingerprint-сигналів, доступних вебресурсам під час формування цифрового відбитка.

Проведене експериментальне дослідження підтвердило коректність роботи реалізованих механізмів. Результати тестування показали, що браузерні профілі використовують визначені конфігурацією параметри цифрової ідентифікації, а локальні дані залишаються ізольованими в межах відповідних Chromium-сесій. Також підтверджено працездатність системи під час взаємодії із зовнішніми сервісами аналізу браузерного відбитка.

Таким чином, поставлена мета роботи досягнута, а всі визначені завдання виконані в повному обсязі. Розроблений програмний засіб IdentityLab може використовуватися для дослідження механізмів цифрової ідентифікації, аналізу браузерних відбитків та вивчення особливостей ізоляції браузерних профілів у сучасному вебсередовищі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Eckersley P. How Unique Is Your Web Browser? // Privacy Enhancing Technologies Symposium. 2010.
2. Laperdrix P., Rudametkin W., Baudry B. Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints // IEEE Symposium on Security and Privacy. 2016.
3. Nikiforakis N., Kapravelos A., Joosen W. et al. Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting // IEEE Symposium on Security and Privacy. 2013.
4. Englehardt S., Narayanan A. Online Tracking: A 1-Million-Site Measurement and Analysis // ACM CCS. 2016.
5. Vastel A. Browser Fingerprinting: A Survey. 2020.
6. Mozilla Foundation. MDN Web Docs. URL: **https://developer.mozilla.org/**
7. Mozilla Foundation. Navigator API. URL: **https://developer.mozilla.org/en-US/docs/Web/API/Navigator**
8. Mozilla Foundation. Canvas API. URL: **https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API**
9. Mozilla Foundation. WebGL API. URL: **https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API**
10. Mozilla Foundation. WebRTC API. URL: **https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API**
11. Mozilla Foundation. Screen API. URL: **https://developer.mozilla.org/en-US/docs/Web/API/Screen**
12. Mozilla Foundation. Web Storage API. URL: **https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API**
13. Mozilla Foundation. IndexedDB API. URL: **https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API**
14. WHATWG. HTML Living Standard. URL: **https://html.spec.whatwg.org/**
15. W3C. Indexed Database API 3.0. URL: **https://www.w3.org/TR/IndexedDB/**

16. W3C. WebRTC 1.0: Real-Time Communication Between Browsers. URL: https://www.w3.org/TR/webrtc/
17. Khronos Group. WebGL Specification. URL: https://www.khronos.org/webgl/
18. Khronos Group. OpenGL ES Specification. URL: https://www.khronos.org/opengles/
19. Electron Team. Electron Documentation. URL: https://www.electronjs.org/docs
20. Electron Team. Session API Documentation. URL: https://www.electronjs.org/docs/latest/api/session
21. Electron Team. WebContents Documentation. URL: https://www.electronjs.org/docs/latest/api/web-contents
22. Electron Team. Webview Tag Documentation. URL: https://www.electronjs.org/docs/latest/api/webview-tag
23. Electron Team. BrowserWindow Documentation. URL: https://www.electronjs.org/docs/latest/api/browser-window
24. Electron Team. Security Recommendations. URL: https://www.electronjs.org/docs/latest/tutorial/security
25. Google. Chromium Project. URL: https://www.chromium.org/
26. Google. Chromium Security Architecture. URL: https://www.chromium.org/Home/chromium-security/
27. Google. Chrome DevTools Protocol. URL: https://chromedevtools.github.io/devtools-protocol/
28. OpenJS Foundation. Node.js Documentation. URL: https://nodejs.org/docs/latest/api/
29. JavaScript.info. The Modern JavaScript Tutorial. URL: https://javascript.info/
30. JSON. Introducing JSON. URL: https://www.json.org/
31. npm Inc. npm Documentation. URL: https://docs.npmjs.com/
32. Multilogin Documentation. URL: https://multilogin.com/
33. AdsPower Documentation. URL: https://www.adspower.com/

34. GoLogin Documentation. URL: https://gologin.com/
35. Dolphin Anty Documentation. URL: https://dolphin-anty.com/
36. Kameleo Documentation. URL: https://kameleo.io/
37. Incogniton Documentation. URL: https://incogniton.com/
38. Regulation (EU) 2016/679 (GDPR). General Data Protection Regulation.
39. Закон України «Про інформацію» від 02.10.1992 №2657-XII.
40. Закон України «Про захист персональних даних» від 01.06.2010 №2297-VI.
41. Закон України «Про електронні довірчі послуги» від 05.10.2017 №2155-VIII.
42. Постанова Кабінету Міністрів України від 17.03.2022 №300 «Деякі питання використання електронних підписів та печаток».