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 004.056:004.738.5(079.2)

ДИПЛОМНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»

Тема: Модуль захисту конфіденційності користувача за допомогою налаштувань відбитка браузера

Виконавець: О.О. Левченко
Керівник: ст. викл. О.В. Дубчак
Нормоконтролер: ст. викл. О.В. Дубчак

Київ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет: Кібербезпеки, комп'ютерної та програмної інженерії

Кафедра: Комп'ютеризованих систем захисту інформації

Освітній ступінь: Бакалавр

Спеціальність: 125 «Кібербезпека»

Освітньо-професійна програма: «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ С.В. Казмірчук

«__» _____ 20__ р.

ЗАВДАННЯ

на виконання дипломної роботи

здобувача вищої освіти

1. Тема: *Модуль захисту конфіденційності користувача за допомогою налаштувань відбитка браузера*

затверджена наказом ректора від «26» квітня 2021 р. № 652/ст.

2. Термін виконання: з 10.05.2021 р. по 20.06.2021 р.

3. Вихідні дані: веб-браузер та його вразливості, існуючі загрози та методи захисту від відбитка браузера, мови програмування JavaScript, HTML, CSS, фреймворк AngularJS, середовище розробки WebStorm, сервіс Panopticlick.

3. Зміст пояснювальної записки: опис і аналіз веб-браузера та його вразливостей, аналіз та дослідження існуючих методів захисту від відбитка браузера; розробка алгоритму та модулю захисту конфіденційності користувача за допомогою налаштувань відбитка браузера, проведення тестування модуля та вироблення рекомендацій щодо його застосування.

КАЛЕНДАРНИЙ ПЛАН

виконання дипломної роботи

№ п/п	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1.	Уточнення постановки задачі	20.04.2021	*Виконано*
2.	Аналіз літературних джерел	21.04.2021 – 03.05.2021	*Виконано*
3.	Обґрунтування вибору рішення	04.05.2021	*Виконано*
4.	Збір інформації	05.05.2021– 10.05.2021	*Виконано*
5.	Опис і аналіз веб-браузера та його вразливостей	11.05.2021 – 13.05.2021	*Виконано*
6.	Аналіз та дослідження методів захисту веб-браузера від відбитка браузера	14.05.2021 – 17.05.2021	*Виконано*
7.	Обрання та опис засобів для реалізації захисту від відбитка браузера	18.05.2021 – 20.05.2021	*Виконано*
8.	Розроблення та тестування модуля захисту конфіденційності користувача за допомогою налаштувань відбитка браузера	21.05.2021 – 01.06.2021	*Виконано*
9.	Вироблення рекомендацій щодо користування розробленим засобом захисту	02.06.2021	*Виконано*
10.	Оформлення презентації	03.06.2021	*Виконано*
11.	Передзахист в ЕК	04.06.2021	*Виконано*
12.	Перевірка на антиплагіат	04.06.2021 – 07.06.2021	*Виконано*
13.	Оформлення і друк пояснювальної записки	08.06.2021 – 13.06.2021	*Виконано*
14.	Отримання рецензії	14.06.2021	*Виконано*
15.	Підготовка до захисту в ЕК	15.06.2021	*Виконано*

Здобувач вищої освіти О.О. Левченко

(підпис, дата)

Керівник дипломної роботи О.В. Дубчак

(підпис, дата)

РЕФЕРАТ

Дипломна робота складається зі вступу, двох розділів, загальних висновків, списку використаних джерел, загальним обсягом робота складає 74 сторінки, має 52 рисунка і 2 таблиці. Список використаних джерел містить 60 найменувань і займає 7 сторінок.

Метою дипломної роботи є створення модуля захисту конфіденційності користувача для браузера Google Chrome задля зменшення кількості конфіденційної інформації, зчитуваної відбитком браузера.

Важливість роботи полягає у проведенні аналізу існуючих вразливостей веб-браузера та засобів боротьби з ними та розробленні пропозиції щодо створення додаткового захисту конфіденційності користувача.

В дипломній роботі проаналізовані існуючі вразливості веб-браузера, одну з головних вразливостей – відбиток браузера, а також існуючі методи захисту від зчитування відбитка браузера. Проаналізовано роботу найпопулярнішого в Україні браузера Google Chrome, на основі знайдених вразливостей в політиці конфіденційності було розроблено авторське розширення для браузера, а також запропоновано рекомендації щодо його використання.

Вперше запропоновано розширення для браузера, яке є комплексним засобом захисту від відбитка браузера і створює додатковий фактор безпеки за рахунок захисту від багатьох складових відбитка браузера, що дає можливість зменшити кількість інформації, зчитуваної відбитком браузера.

Дане розширення є готовим продуктом, який може використовуватися як рядовими користувачами, так і комерційними організаціями. Розширення дозволить убезпечити комп’ютерну систему від цільової реклами та попередити зловмисницькі дії щодо витоку конфіденційної інформації.

Ключові слова: веб-браузер, відбиток браузера, інформаційна безпека, загроза, захист, розширення, кібербезпека, конфіденційність.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 6

ВСТУП 7

РОЗДІЛ 1. ВЕБ-БРАУЗЕР – ОБ'ЄКТ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ 9

1.1. Основна термінологія кібербезпеки 9
1.2. Веб-браузер – кіберпростір під загрозою атак 11
1.2.1. Браузер, його функції та призначення. 13
1.2.2. Особливості браузера та пов'язані з ними вразливості. 14
1.3. Відбиток браузера як одна з найголовніших загроз безпеці конфіденційності користувача 19
1.3.1. Складові відбитка браузера. 22
1.3.2. Існуючі методи боротьби з відбитком браузера. 26
1.3.3. Захист від відбитка браузера на законодавчому рівні. 33
1.4. Висновки до розділу 1 34

РОЗДІЛ 2. РОЗРОБКА ВЛАСНОГО МОДУЛЮ ЗАХИСТУ ВІД ВІДБИТКА БРАУЗЕРА 36

2.1. Браузер Google Chrome як об'єкт захисту від відбитка браузера, засоби вбудованого захисту в браузері Google Chrome 37
2.2. Авторський варіант захисту від відбитка браузера 42
2.2.1. Середовище розробки програмного додатку WebStorm. 43
2.2.2. Фреймворк AngularJS. 43
2.2.3. Мови розробки. 44
2.2.4. Сервіс Panopticlick. 45
2.2.5. Створення розширення «ПарашутOff» для браузера 45
2.2.6. Рекомендації щодо використання розширення 52
2.2.7. Тестування створеного розширення на предмет ефективності захисту від відбитка браузера. 58
2.4. Висновки до розділу 2 63

ВИСНОВКИ 65

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 68

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API	– Application Programming Interface	– прикладний програмний інтерфейс;
CPU	– Central Processing Unit	– центральний процесор;
CSS	– Cascading Style Sheets	– каскадні таблиці стилів;
DNT	– Do Not Track	– не відслідковувати;
HTML	– HyperText Markup Language	– мова розмітки гіпертексту;
HTTP	– Hyper Text Transfer Protocol	– протокол передачі гіпертекстових документів;
HTTPS	– Hyper Text Transfer Protocol Secure	– захищений протокол передачі гіпертекстових документів;
IP	– Internet Protocol	– протокол Інтернета;
JVM	– Java Virtual Machine	– віртуальна машина Java;
RTC	– Real-Time Communications	– комунікація в реальному часі;
URL	– Uniform Resource Locator	– уніфікований локатор ресурсів;
VPN	– Virtual Private Network	– віртуальна приватна мережа;
WebGL	– Web Graphics Library	– бібліотека веб-графіки;
XSS	– Cross-Site Scripting	– міжсайтовий скриптинг;
ВМ	– віртуальна машина;	
НСД	– несанкціонований доступ;	
ОЗП	– оперативна пам'ять;	
ОС	– операційна система;	
ПЗ	– програмне забезпечення.	

ВСТУП

Актуальність. Інтернетом користуються 4,6 мільярда людей [1], і їх безпека напряму залежить від програмного забезпечення (ПЗ), що використовується для доступу до веб-сайтів та їх перегляду – веб-браузера [2].

Більшість людей користується браузерами, але мало хто знає, як уникнути зчитування персональних даних, в зв'язку з чим кількість унікальних кіберінцидентів, які використовують вразливості браузера, зросла зі 141 в 2019 р. до 223 в 2020 р. [3]. З них 65% були спрямовані на отримання даних [3]. Майже третина всіх веб-браузерів містить критичні вразливості [4].

На сьогоднішній день гарантовано дієвих методів і засобів захисту від технології зчитування відбитка браузера не розроблено [5], отже необхідним є дослідження щодо підвищення захисту конфіденційності користувача при роботі з веб-браузером.

Оцінка сучасного стану проблеми на основі вітчизняної та зарубіжної літератури. Серед науковців та практиків, які проводять дослідження в напрямку вразливостей браузера, зокрема відбитка браузера, переважать зарубіжні, серед яких Іньчжи Цао (Yinzhi Cao) [6], Сонг Лі (Song Li) [6], Ерік Війманс (Erik Wijmans) [6], Пітер Еккерслі (Peter Eckersley) [7].

Метою дипломної роботи є створення модуля захисту конфіденційності користувача для браузера Google Chrome задля зменшення кількості конфіденційної інформації, зчитуваної відбитком браузера.

Досягнення мети потребує розв'язання таких **завдань**:

– опис і аналіз веб-браузера та його вразливостей;

– аналіз та дослідження існуючих методів захисту від відбитка браузера;

– розробка модуля захисту конфіденційності користувача за допомогою налаштувань відбитка браузера та проведення його тестування;

– формування практичних рекомендацій щодо використання розробленого модуля.

Об'єкт дослідження: процес захисту конфіденційності користувача інформаційно-комунікаційною системою.

Предмет дослідження: методи та засоби захисту конфіденційності користувача.

Методи дослідження:

- порівняльно-аналітичний метод щодо існуючих загроз веб-браузеру та наявних засобів захисту від відбитка браузера;
- метод об’єктно-орієнтованого програмування для створення модуля захисту конфіденційності користувача.

Галузь застосування. Дане розширення є готовим продуктом, який може використовуватися як рядовими користувачами, так і комерційними організаціями. Розширення дозволить убезпечити комп’ютерну систему від цільової реклами та попередити зловмисницькі дії щодо витоку конфіденційної інформації.

Новизна. Вперше запропоновано розширення для браузера, яке є комплексним засобом захисту від відбитка браузера і створює додатковий фактор безпеки за рахунок захисту від багатьох складових відбитка браузера, що дає можливість зменшити кількість інформації, зчитуваної відбитком браузера.

Практична цінність роботи полягає у проведенні детального аналізу існуючих вразливостей веб-браузера, а також наявних засобів захисту від відбитка браузера. Розроблено авторське розширення, яке створює додатковий фактор безпеки за рахунок захисту від багатьох складових відбитка браузера, що дає можливість зменшити кількість інформації, зчитуваної відбитком браузера. Для реалізації використано мови програмування JavaScript, HTML, CSS, фреймворк AngularJS та середовище розробки WebStorm. Дане розширення є готовим продуктом, який може використовуватися як рядовими користувачами, так і комерційними організаціями.

РОЗДІЛ 1. ВЕБ-БРАУЗЕР – ОБ'ЄКТ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Зараз суспільство живе в епоху, коли використання Інтернету стало цілком природним для мільярдів людей [1]. У сучасних цифрових умовах на цю мережу покладається багато щоденної діяльності. Різні форми спілкування, розваги, а також фінансові та робочі завдання виконуються в Інтернеті, що означає постійну передачу великих обсягів даних та конфіденційної інформації. Така залежність від Інтернету приносить нові та небезпечні ризики.

Як відомо, веб-браузери є своєрідними «воротами», відкриваючими доступ до Інтернету [2], отже безпека веб-браузера є важливим критерієм захисту даних користувача, до яких зловмисники хочуть отримати доступ.

1.1. Основна термінологія кібербезпеки

Закон України «Про основні засади забезпечення кібербезпеки України» [8] дає такі визначення основним поняттям:

1) інцидент кібербезпеки (кіберінцидент) – подія або ряд несприятливих подій ненавмисного характеру (природного, технічного, технологічного, помилкового, у тому числі внаслідок дії людського фактора) та/або таких, що мають ознаки можливої (потенційної) кібератаки, які становлять загрозу безпеці систем електронних комунікацій, систем управління технологічними процесами, створюють імовірність порушення штатного режиму функціонування таких систем (у тому числі зриву та/або блокування роботи системи, та/або несанкціонованого управління її ресурсами), ставлять під загрозу безпеку (захищеність) електронних інформаційних ресурсів; [8]

2) кібератака – спрямовані (навмисні) дії в кіберпросторі, які здійснюються за допомогою засобів електронних комунікацій (включаючи інформаційно-комунікаційні технології, програмні, програмно-апаратні засоби, інші технічні та технологічні засоби і обладнання) та спрямовані на досягнення однієї

або сукупності таких цілей: порушення конфіденційності, цілісності, доступності електронних інформаційних ресурсів, що обробляються (передаються, зберігаються) в комунікаційних та/або технологічних системах, отримання несанкціонованого доступу (НСД) до таких ресурсів; порушення безпеки, надійного та штатного режиму функціонування комунікаційних та/або технологічних систем; використання комунікаційної системи, її ресурсів та засобів електронних комунікацій для здійснення кібератак на інші об'єкти кіберзахисту; [8]

3) кібербезпека – захищеність життєво важливих інтересів людини і громадянина, суспільства та держави під час використання кіберпростору, за якої забезпечуються сталий розвиток інформаційного суспільства та цифрового комунікативного середовища, своєчасне виявлення, запобігання і нейтралізація реальних і потенційних загроз національній безпеці України; [8]

4) кіберзагроза – наявні та потенційно можливі явища і чинники, що створюють небезпеку життєво важливим національним інтересам України у кіберпросторі, справляють негативний вплив на стан кібербезпеки держави, кібербезпеку та кіберзахист її об'єктів; [8]

5) кіберзахист – сукупність організаційних, правових, інженерно-технічних заходів, а також заходів криптографічного та технічного захисту інформації, спрямованих на запобігання кіберінцидентам, виявлення та захист від кібератак, ліквідацію їх наслідків, відновлення сталості і надійності функціонування комунікаційних, технологічних систем; [8]

6) кіберзлочин (комп'ютерний злочин) – суспільно небезпечне винне діяння у кіберпросторі та/або з його використанням, відповідальність за яке передбачена законом України про кримінальну відповідальність та/або яке визнано злочином міжнародними договорами України; [8]

7) кіберзлочинність – сукупність кіберзлочинів; [8]

8) кібероборона – сукупність політичних, економічних, соціальних, військових, наукових, науково-технічних, інформаційних, правових, організаційних та інших заходів, які здійснюються в кіберпросторі та спрямовані на за-

безпечення захисту суверенітету та обороноздатності держави, запобігання виникненню збройного конфлікту та відсіч збройній агресії; [8]

9) кіберпростір – середовище (віртуальний простір), яке надає можливості для здійснення комунікацій та/або реалізації суспільних відносин, утворене в результаті функціонування сумісних (з'єднаних) комунікаційних систем та забезпечення електронних комунікацій з використанням мережі Інтернет та/або інших глобальних мереж передачі даних. [8]

1.2. Веб-браузер – кіберпростір під загрозою атак

Сьогодні такі веб-браузери, як Google Chrome, Opera, Microsoft Internet Explorer, Mozilla Firefox та Apple Safari встановлені майже на всіх комп'ютерах [2]. Оскільки веб-браузери використовуються так часто, важливо налаштувати їх безпеку, але в переважній більшості веб-браузери, що постачаються з операційною системою (ОС), за замовчуванням не мають таких налаштувань [9]. Якщо не захистити свій веб-браузер, це може швидко призвести до різноманітних комп'ютерних проблем, спричинених будь-чим: від встановлення шпигунського ПЗ без відома користувача до зловмисників, які контролюють комп'ютер жертви.

В ідеалі, користувачі комп'ютерів повинні оцінювати ризики ПЗ, яке вони використовують. Багато комп'ютерів продаються з уже завантаженим ПЗ [9]. Незалежно від того, встановлено воно виробником комп'ютера, виробником ОС, постачальником послуг Інтернету або роздрібним магазином, першим кроком у аналізі вразливості комп'ютера є оцінювання встановленого ПЗ та взаємодії програм між собою. На жаль, більшість людей не вважають за потрібне проводити такий рівень аналізу [10].

Зростає загроза атак, які використовують вразливості веб-браузерів [3]. Вони потрапляють до веб-браузерів через використання скомпрометованих або

шкідливих веб-сайтів [11]. Цю проблему погіршує низка факторів, зокрема такі як:

- багато користувачів схильні натискати на посилання, не беручи до уваги ризики своїх дій; [11]
- адреси веб-сторінок можуть бути замасковані або переносять на несподіваний сайт; [11]
- багато веб-браузерів налаштовані на забезпечення розширених функціональних можливостей за рахунок зниження рівня безпеки; [11]
- комп’ютерні системи та програмні пакети можуть комплектуватися додатковим ПЗ, що збільшує кількість вразливих місць, які атакуються; [11]
- багато веб-сайтів вимагають, щоб користувачі вмикали певні функції або встановлювали більше ПЗ, що створює додатковий ризик для комп'ютера; [11]
- багато користувачів не знають, як безпечно налаштувати свої веб-браузери; [11]
- багато користувачів не бажають вмикати або вимикати функціональні можливості, необхідні для захисту веб-браузера. [11]

Як результат, використання вразливостей веб-браузерів стало популярним способом зловмисників компрометувати комп’ютерні системи (рис. 1.1).

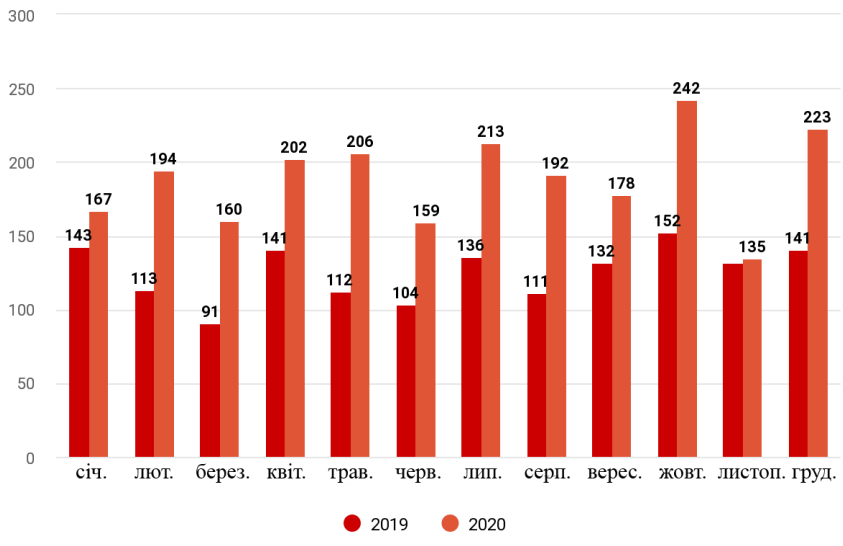


Рис. 1.1. Кількість інцидентів в 2019 і 2020 роках [3]

1.2.1. Браузер, його функції та призначення.

Веб-браузер або просто «браузер» – це ПЗ, що використовується для доступу до веб-сайтів та їх перегляду. [2]

Основною функцією веб-браузера є візуалізація HTML-коду (HyperText Markup Language – мова розмітки гіпертексту [12]), що використовується для дизайну або «розмітки» веб-сторінок. Кожного разу, коли браузер завантажує веб-сторінку, він обробляє HTML-код, який може включати текст, посилання, зображення та інші елементи, такі як функції CSS (Cascading Style Sheets, каскадні таблиці стилів [12]) та JavaScript. Браузер обробляє ці елементи, а потім відтворює їх. [2]

Призначення веб-браузера – отримати вміст з Інтернету та відобразити його на пристрої користувача. Цей процес починається, коли користувач вводить у браузер URL (Uniform Resource Locator, уніфікований локатор ресурсів [13]). Практично всі URL-адреси в Інтернеті починаються з http: або https: що означає, що браузер отримує їх за допомогою HTTP (Hyper Text Transfer Protocol, протокол передачі гіпертекстових документів [14]). У випадку https: (Hyper Text Transfer Protocol Secure, захищений протокол передачі гіпертекстових документів [14]) зв'язок між браузером та веб-сервером зашифрований з метою безпеки та конфіденційності. [15]

Після отримання веб-сторінки механізм візуалізації браузера відображає її на пристрої користувача. Сюди входять формати зображень та відео, що підтримуються браузером. [15]

Веб-сторінки зазвичай містять гіперпосилання на інші сторінки та ресурси. Кожне посилання містить URL-адресу, і при натисканні браузер переходить до нового ресурсу [2]. Таким чином, процес доведення вмісту до користувача починається знову.

Ранні веб-браузери, такі як Mosaic та Netscape Navigator, були простими програмами, які візуалізували HTML-код, обробляли введення форм та підтримували закладки. У міру того, як веб-сайти еволюціонували, змінювалися і вимоги до веб-браузера. Сучасні браузери набагато вдосконаленіші, вони підтри-

мують декілька типів HTML, динамічний JavaScript та шифрування, що використовуються захищеними веб-сайтами. [16]

Сучасні веб-браузери дають можливість веб-розробникам створювати високоінтерактивні веб-сайти. Наприклад, Ajax дозволяє браузеру динамічно оновлювати інформацію на веб-сторінці без необхідності перезавантажувати сторінку. Досягнення CSS дозволяють браузерам відображати адаптивні макети веб-сайтів та широкий спектр візуальних ефектів. [16]

У світі станом на 2021 рік найпоширенішим браузером є Google Chrome, на другому місці Apple Safari, далі Edge, Mozilla Firefox, та Opera (рис. 1.2). [17]

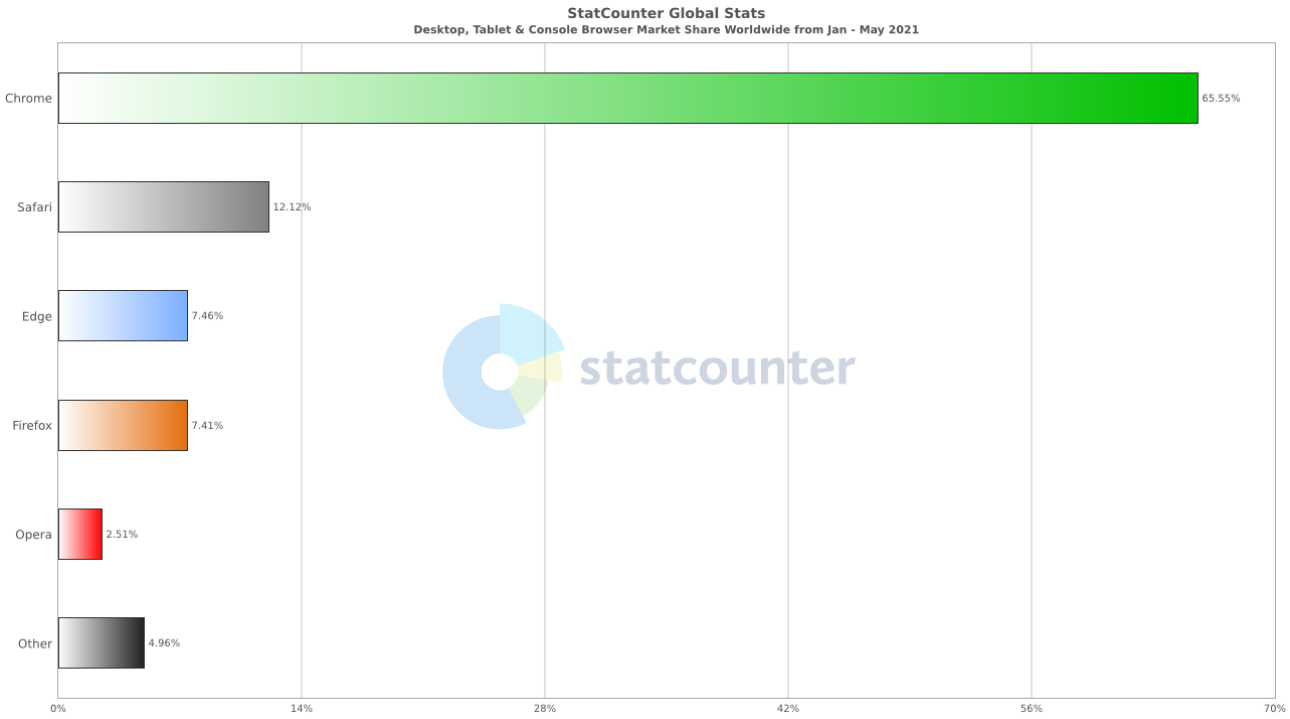


Рис. 1.2. Найпопулярніші браузери станом на 2021 рік [17]

1.2.2. Особливості браузера та пов’язані з ними вразливості.

Користувачу важливо розуміти функціональність та можливості веб-браузера, яким він користується. Увімкнення деяких функцій веб-браузера може знизити рівень безпеки. Постачальники часто вмикають функції за замовчуванням для полегшення роботи з комп’ютером, але ці функції можуть в підсумку збільшити ризик для комп’ютера. [9]

Конкретні особливості веб-браузера та пов'язані з ними ризики наведені та описані нижче (рис. 1.3). Знання того, що роблять різні функції, допоможе користувачу зрозуміти, як вони впливають на функціональність веб-браузера та безпеку комп’ютера.

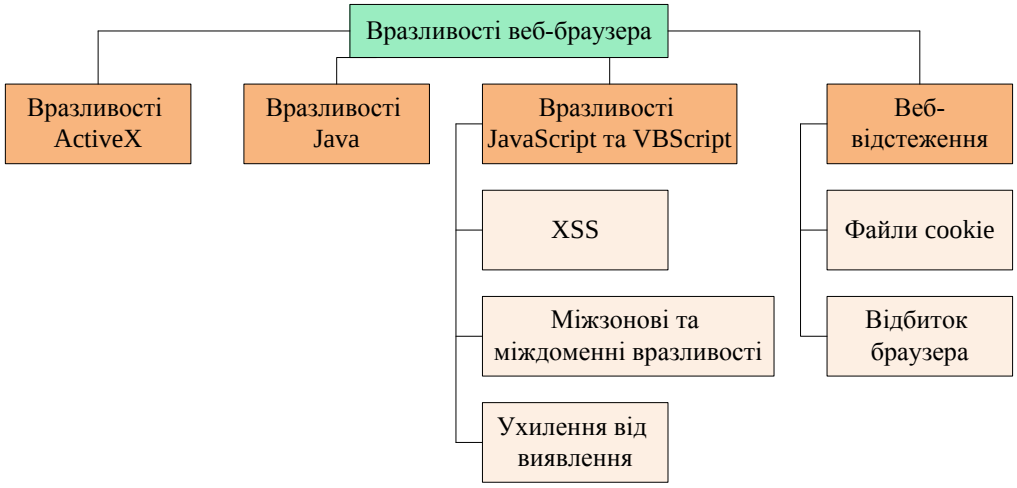


Рис. 1.3. Вразливості веб-браузера

ActiveX – це технологія, що використовується Microsoft Internet Explorer у системах Microsoft Windows. ActiveX дозволяє веб-браузеру використовувати програми або їх частини. Веб-сторінка може використовувати компоненти ActiveX, які вже знаходяться в системі Windows, або надавати компонент як завантажуваний об’єкт. Це дає додаткову функціональність традиційному перегляду веб-сторінок, але може також створити більш серйозні вразливості. [18]

Багато вразливостей елементів керування ActiveX призводять до серйозних наслідків. Так, наприклад, скориставшись вразливістю в елементі ActiveX – NBBDownloader.ocx – хакер може створити спеціальну веб-сторінку, при відвідуванні якої користувачем стає можливим віддалене виконання коду. Зловмисник, що успішно скористався цією вразливістю, може отримати ті ж права, якими володіє локальний користувач [19]. Такий сценарій використання вразливостей, пов’язаних з ActiveX, доволі часто зустрічається в базі даних вразливостей [20].

Java – це об’єктно-орієнтована мова програмування, яку можна використовувати для розробки активного вмісту веб-сайтів [21]. JVM (Java Virtual

Machine, віртуальна машина Java [21]) використовується для запуску коду Java, або «Java-аплету» [21], наданого веб-сайтом.

Аплети Java зазвичай виконуються в межах «пісочниці», де взаємодія з рештою системи обмежена [22]. Однак різні реалізації JVM містять вразливості, які дозволяють аплетам обходити ці обмеження. Підписані аплети Java також можуть обходити обмеження пісочниці, але вони, як правило, спочатку запитують у користувача право на виконання. [21]

Найпоширенішою вразливістю, що використовує аплети Java [23], є вразливість продукту Java SE (Standard Edition) від Oracle Java SE. Вона дозволяє неавторизованому зловмиснику з мережевим доступом через кілька протоколів компрометувати Java SE. Успішні атаки цієї вразливості можуть призвести до НСД для читання даних, доступних для Java SE. Ця вразливість стосується клієнтів, що працюють із захищеними програмами Java Web Start або ізольованими аплетами Java, які завантажують та запускають ненадійний код (наприклад, код, що надходить з Інтернету), а для забезпечення покладаються на пісочницю Java. [24]

JavaScript, також відомий як ECMAScript, – це мова сценаріїв, яка використовується для того, щоб зробити веб-сайти більш інтерактивними. У стандарті JavaScript є специфікації, які обмежують певні функції, такі як доступ до локальних файлів. [25]

VBScript (Visual Basic Script) – інша мова сценаріїв, унікальна для Microsoft Windows Internet Explorer. VBScript схожий на JavaScript, але він не настільки широко використовується на веб-сайтах через обмежену сумісність з іншими браузерами. [26]

Можливість запуску мови сценаріїв, таких як JavaScript або VBScript, дозволяє авторам веб-сторінок додавати значну кількість функцій та інтерактивності до веб-сторінки. Однак цією ж можливістю зловмисники можуть зловживати. Конфігурація за замовчуванням для більшості веб-браузерів включає підтримку сценаріїв, які можуть вводити численні вразливості, наприклад, такі:

- **XSS** (Cross-Site Scripting, міжсайтовий скриптинг) – це вразливість веб-сайту, що дозволяє кіберзлочинцю використовувати довірчі відносини, які користувач має з цим сайтом. Наприклад, якщо зловмисник підробить сторінку авторизації сайту, змінивши всього одну літеру в його адресі, то зможе отримати логін і пароль жертви. [27]
- **Міжзонові та міждоменні вразливості**. Більшість веб-браузерів використовують моделі безпеки, щоб запобігти доступу скриптів на веб-сайті до даних в іншому домені [28]. Ці моделі безпеки в основному базуються на політиці того ж походження Netscape [29].

Вразливості, що порушують ці моделі безпеки, використовуються для виконання дій, які сайт зазвичай не може виконувати. Якщо вразливість дозволяє зловмиснику перейти в локальну машинну зону або інші захищені зони, він може виконувати які завгодно команди у вразливій системі. [30]

- **Ухилення від виявлення**. Зазвичай антивірусні системи, системи виявлення вторгнень та системи запобігання вторгненню працюють, шукаючи конкретні закономірності у вмісті. Якщо виявлено «поганий» шаблон, який є в базах, тоді будуть вжиті відповідні дії для захисту користувача. Однак через динамічну природу мов програмування сценарії на веб-сторінках можуть використовуватися для уникнення таких захисних систем. [28]

Веб-відстеження – це дискусійний прийом, який використовується для запам’ятовування і розпізнавання відвідувачів веб-сайту. З одного боку, веб-відстеження може автентифікувати користувачів (особливо корисною є комбінація різних методів веб-відстеження для багатофакторної автентифікації задля посилення безпеки). З іншого боку, веб-відстеження також може використовуватися для надання персоналізованих послуг – і якщо послуга небажана (небажана цільова реклама), таке відстеження є порушенням конфіденційності. [6]

Веб-відстеження швидко розвивається. Техніка відстеження першого покоління приймає встановлені сервером ідентифікатори, такі як файли cookie та evercookie. Техніка відстеження другого покоління, яка називається відбитком браузера, ідентифікує пристрій на основі його унікальних конфігурацій, таких

як часовий пояс, системні шрифти, модулі до браузеру і їх версії, журнал відвідувань, розширення екрану. [6]

Файли cookie – це файли, розміщені в системі користувача для зберігання даних для певних веб-сайтів. Вони можуть містити будь-яку інформацію: від відомостей про відвідані веб-сайти до даних для доступу до веб-сайту. Файли cookie призначені для читання лише веб-сайтами, що створили файли cookie. Сеансові файли cookie видаляються після закриття браузера, а постійні файли cookie залишатимуться на комп'ютері до зазначеної дати закінчення. [31]

Файли cookie можуть використовуватися для унікальної ідентифікації відвідувачів веб-сайту, що деякі люди вважають порушенням конфіденційності. Якщо веб-сайт використовує файли cookie для автентифікації, зловмисник може отримати НСД до цього сайту, викравши файл cookie. Постійні файли cookie становлять більший ризик, ніж сеансові, оскільки вони довше залишаються на комп'ютері. [31]

Технологія cookie вважається одним з головних засобів, які власники інтернет-ресурсів використовують для відстеження клієнтів їх ресурсу. Однак ця методика поступово застаріває і нерідко не дає необхідний ефект. [31]

Цьому сприяють декілька причин. У сучасних реаліях практично будь-який користувач Інтернету може деактивувати операцію отримання cookie або скористуватися вбудованим в веб-браузер режимом «інкогніто» і зберегти cookie тільки на поточну сесію. Ці прийоми дозволяють зробити присутність користувача і його дії для сайту непоміченими. [31]

Ситуація з технологією **відбитка браузера** в корені відрізняється. Ця технологія базується на аналізі інформації, отриманої від браузера клієнта при відвідуванні електронного ресурсу. Завдяки цим даним створюється цілісна картина браузера, яка схожа за своїм принципом відображення з відбитком пальця. В результаті навіть при відключенні або видаленні cookie ресурс все одно пізнає конкретного користувача по його відбитку веб-браузера. [32]

Відбитки, що ідентифікують браузер користувача, замінюють собою cookie. Важливо розуміти, що налаштування браузера з метою захиститися від

надмірної активності сайтів по ідентифікації користувачів може привести до того, що такі налаштування роблять браузер більш впізнаваним. [32]

1.3. Відбиток браузера як одна з найголовніших загроз безпеці конфіденційності користувача

Дослідники Інтернет-безпеки у 2020 році підготували звіт про стан сучасних веб-браузерів на основі дослідження використання провідних браузерів 1,4 мільйонами користувачів у всьому світі. [4]

Основний результат полягає в тому, що майже третина всіх веб-браузерів містить критичні вразливості: [4]

– Microsoft Internet Explorer і Edge – понад 40%;
– Google Chrome – трохи менше 40%;
– Mozilla Firefox – 35%;
– Opera – 34%;
– Safari менше – 30%.

Серед вразливостей найбільший відсоток займають ті, що спрямовані на отримання даних (рис. 1.4).

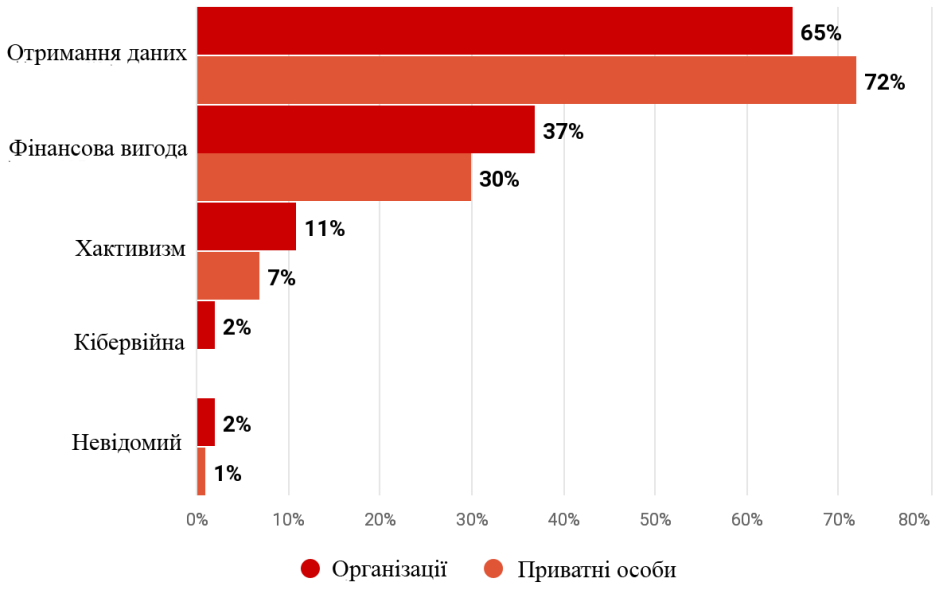


Рис. 1.4. Мотиви зловмисників (частка атак) [3]

Отже, загроза порушення конфіденційності – основна причина, через яку користувач повинен бути уважний. Зважаючи на це, технологія відбитка браузера багато в чому небезпечніше інших вразливостей. До того ж, від неї складніше захиститися, так як неможливо дізнатися про те, стежать за користувачем чи ні. Система позначає персональний комп'ютер відвідувача унікальною цифровою міткою в вигляді хеш-суми, знятої за особливим алгоритмом з налаштувань браузера, про присутність якої користувач навіть не здогадується. Таким чином, створюється база міток для ідентифікації користувачів. При наступному відвідуванні користувачем ресурсу проводиться порівняння відбитка його браузера з базою міток і при збігу відбувається однозначна ідентифікація. [32]

Технологія відбитка браузера є глобальним ідентифікатором, що робить його власника більш впізнаваним не тільки на часто відвідуваних інтернет-ресурсах, а й в інших електронних джерелах. Відбиток браузера фіксує цілісну картину, яку ресурс отримує від веб-браузера. Це дозволяє зробити ідентифікацію клієнта навіть при внесенні змін до налаштувань браузера. Як доказ цього, на рис. 1.5 видно, що за 24 години менше 10% неодноразово спостережуваних пристроїв зуміли змінити свої цифрові відбитки. При цьому інформація про налаштування безпеки, які наче повинні давати більш надійний захист, теж входить до відбитка браузера. [33]

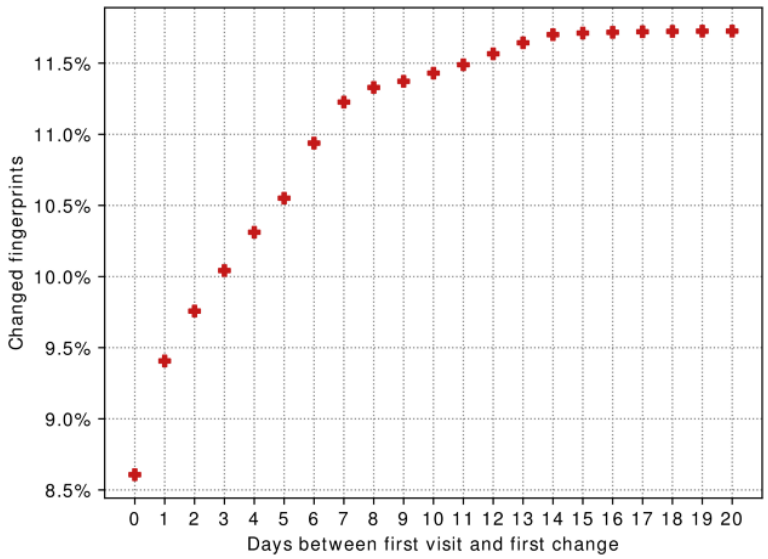


Рис. 1.5. Швидкість зміни відбитка браузера [33]

Значну загрозу становить використання «цифрового двійника» – це вид шахрайства, що базується на відбитку браузера. У даркнеті існує ринок під назвою «Genesis», де продаються цифрові маски, які включають в себе історію відвідин сайтів, інформацію про ОС, браузери, і так далі. Тобто інформацію, яку збирає відбиток браузера. [34]

Таким чином, при наявності цифрової маски і облікових даних користувача в системі, система захисту може прийняти інтернет-шахрая за легітимного користувача і не стане вживати які-небудь заходи протидії. Саме з цієї причини шахраї збирають всі можливі дані з пристроїв, а потім продають їх на «Genesis». Придбавши ці дані, зловмисники видають себе за власників цифрової маски. [34]

Відбиток браузера може бути кращим другом хакера. Якщо зловмиснику відомі точні дані про пристрій жертви, він може використовувати спеціальні експлойти (програми, що використовують вразливості ПЗ для проведення атаки) для його злому. В цьому немає нічого складного – будь-який кіберзлочинець може створити підроблений сайт зі скриптом зняття відбитків (рис. 1.6). Знання вразливостей і особливостей пристрою користувача полегшує будь-яку атаку, в чому кіберзлочинцям і допомагають відбитки браузера. [35]

Рис. 1.6. Використання порушником інформації про відбиток браузера

1.3.1. Складові відбитка браузера.

Відбитки браузера являють собою унікальні значення, що відображають налаштування браузера та комп'ютера користувача. Нижче представлені характеристики, що збираються відбитком браузера. Графічне їх представлення можна побачити на рис. 1.7.

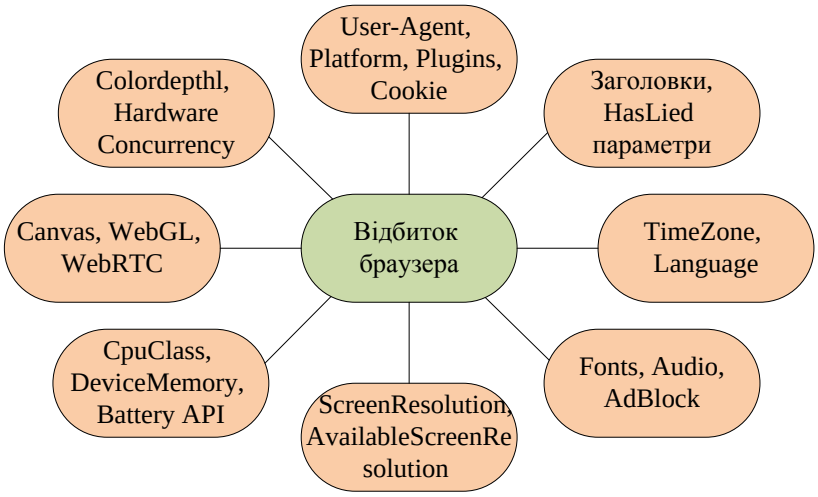


Рис. 1.7. Складові відбитка браузера

- User-Agent – унікальний рядок-ідентифікатор [36]. Нижче наведено приклад розбору рядка User-Agent: «Mozilla/5.0 (X11; Linux x86_64; en-US; rv:57.0) Gecko/20100101 Firefox/57.0»:
  - «Mozilla/5.0»: оригінальне кодове ім'я Navigator;
  - «X11; Linux x86_64»: ОС і апаратна платформа комп'ютера;
  - «en-US»: мова локалізації;
  - «rv:57.0»: версія верстки;
  - «Gecko/20100101»: кодове найменування ПЗ, що перетворює вміст веб-сторінок і інформацію про форматування в інтерактивне зображення форматованого вмісту на екрані / збірка;
  - «Firefox/57.0»: ім'я браузера і версія.
- TimeZone – назва часового поясу і часовий пояс, який визначається числовим параметром зсуву щодо нульової зони. Ця метрика може бути вико-

ристана для визначення загального місцезнаходження користувача, особливо якщо він живе в часовому поясі, де мало інших користувачів. [36]

- ScreenResolution – роздільна здатність екрану – довжина і ширина екрану в пікселях. За даними цієї характеристики обчислюється параметр «Діагональ екрану». Варто зазначити, що при переведенні пікселів в дюйми частина інформації втрачається. У зв'язку з тим, що новий ідентифікатор представляється одним числом, в деяких випадках такий перерозподіл може бути ефективним. [36]
- Canvas – унікальний рядок-ідентифікатор, який утворюється шляхом конвертації градієнтного кольорового об'єкта за допомогою кодування base64. Унікальність досягається за рахунок того, що різні браузери, ОС та графічні відеокарти мають невеликі відмінності в відображенні одного і того ж елемента, тому в 90% випадках вас можна визначити навіть після зміни IP (Internet Protocol, протокол Інтернета). Дана ознака характеризується низькою ймовірністю повторюваності і колізій в завданні ідентифікації. [36]
- WebGL (Web Graphics Library, бібліотека веб-графіки) – унікальний рядок-ідентифікатор, який отримують конвертацією градієнтного об'єкта за допомогою кодування base64 з урахуванням усіх винятків і можливостей WebGL. Метод створення хешу відбитка WebGL дуже схожий на створення хешу відбитка Canvas. Результати зчитування відбитків WebGL та Canvas тісно пов’язані. Вони обидва досліджують графіку, відтворену браузером, на предмет незначних відмінностей між користувачами. [36]
- DeviceMemory – розмір оперативної пам'яті (ОЗП) на пристрої. Буде корисним в поєднанні з іншими показниками, але не є самостійним ідентифікатором. [36]
- HardwareConcurrency – максимально можливе задіяне число потоків на комп'ютері. Буде корисним в поєднанні з іншими показниками, але не є самостійним ідентифікатором. [36]

- AdBlock – параметр, що визначає факт працездатність модуля AdBlock (блокування реклами). Додає мало інформації до відбитка браузера, але все-таки може бути мінімально корисним для ідентифікації. [36]
- TouchSupport – параметр, що визначає характеристики touchscreen. Цей показник стосується кількості точок дотику на пристрої, такому як планшет або телефон. [36]
- Language – мова користувацького інтерфейсу браузера. Особливо корисний показник, якщо мова є незвичною для часового поясу. [36]
- Colordepth – число біт, що визначають глибину кольору для одного пікселя. Додає мало інформації до відбитка браузера, але все-таки може бути мінімально корисним, щоб ідентифікувати користувача. [36]
- AvailableScreenResolution – доступна роздільна здатність екрану для вікна. Хоча цей показник може доповнювати іншу інформацію, він часто є занадто «крихким», щоб використовувати його окремо, оскільки користувачі можуть легко змінити розміри вікна свого браузера. [36]
- SessionStorage – параметр, що дозволяє визначити можливість збереження даних в сесійному сховищі. [36]
- LocalStorage – параметр, що дозволяє визначити можливість збереження даних в локальному сховищі. [36]
- AddBehavior – параметр, що дозволяє визначити можливість використання поведінкових ознак. [36]
- CpuClass (Central Processing Unit, центральний процесор, ЦП) – клас ЦП ОС користувача. Цей показник може виявитися корисним для ідентифікації. Особливо це стосується систем із спеціальним обладнанням. [36]
- Platform – назва платформи браузера, яка представляє прямий спосіб взаємодії додатків користувача з ОС Windows. Унікальність цього параметра залежить від машини користувача. [36]
- Plugins – масив відомостей про плагіни, встановлені в додатку. Плагін являє собою самостійний програмний модуль, що підключається до основ-

ної програми і призначений для розширення використовуваних можливостей. Даний параметр включає найменування плагінів, їх опис і типи. Актуально для старих браузерів, оскільки сучасні віддають перевагу більш регламентованим доповненням і розширенням. [36]

- HasLiedLanguage – параметр, який перевіряє факт збігу мови призначеного для користувача інтерфейсу браузера з першою мовою в списку найбільш бажаних мов користувача. [36]
- HasLiedResolution – параметр, який перевіряє факт збігу роздільної здатності екрану з доступною роздільною здатністю екрану. [36]
- HasLiedBrowser – параметр, який перевіряє факт відповідності між даними про браузер, витягнутими з User-Agent, і даними про те, як браузер справляється з штучно створеною помилкою. [36]
- Fonts – список шрифтів, доступних у браузері. [36]
- Audio – число, що відображає суму буферних значень. Для отримання даного параметра веб-сайт відправляє запит браузеру на моделювання синусоїда, заснованого на результатах аудіо-стека пристрою. Потім результат відправляється на сервер і застосовується аналогічно з ентропією для унікальної ідентифікації. Як і відбитки Canvas, цей ідентифікатор може бути унікальним залежно від звукової карти та драйверів і, як правило, не змінюється з часом. Для настільних комп'ютерів, особливо тих, що мають спеціальне обладнання, звукова карта надасть нову інформацію. [36]
- HTTP_accept headers – веб-заголовок, який використовується, щоб повідомити серверу, які типи вмісту може обробляти браузер. Наприклад, сервер може вибрати відображення простого текстового файлу, якщо побачить, що браузер користувача не підтримує розширені документи. Ця інформація може бути досить унікальною і варіюється від браузера до браузера, не змінюється із часом і залишається тією самою у багатьох версіях одного браузера. [36]
- Cookie – параметр, що зазначає, увімкнено дозвіл на зберігання файлів cookie чи ні. [36]

- DNTHeader (Do Not Track, не відслідковувати) – веб-заголовок, який використовується, щоб повідомити сервер, що користувач вважає за краще не відстежуватись. На жаль, більшість сайтів ігнорують цей заголовок. [36]
- WebRTC (Real-Time Communications – комунікація в реальному часі) – інформація про локальну IP-адресу пристрою користувача. [36]
- Battery API (Application Programming Interface, прикладний програмний інтерфейс) – містить дані про відсоток заряду акумулятора, стан зарядки, і з цього можна зробити висновок, що використовує людина: ноутбук, телефон чи планшет. [36]
- HTTP «Referer» – веб-заголовок, що повідомляє веб-сайтам URL-адресу, яка спрямовувала користувача на дану сторінку. Реферальний журнал використовується, щоб веб-сайти та веб-сервери могли визначати, звідки люди їх відвідують, для рекламних або статистичних цілей. [36]

1.3.2. Існуючі методи боротьби з відбитком браузера.

На сьогоднішній день гарантовано дієвих методів і засобів захисту від технології зчитування відбитка браузера поки не розроблено, однак є заходи, застосування яких дозволяє радикально знизити унікальність веб-браузера. [5]

Тут допоможуть такі інструменти:

- **Зміна налаштувань браузера.** Найочевидніший засіб, проте малоефективний і недовготривалий. Сюди відноситься модернізація браузерів, використання типових плагінів, зміна роздільної здатності екрану під найбільш поширену, видалення нестандартних шрифтів, зміна часового поясу. Тобто усі налаштування повинні зробити браузер якомога менш унікальним. [5]
- **Браузер Firefox з модифікованими налаштуваннями.** Цей браузер непоганий в питанні захисту призначених для користувача даних. Нещодавно розробники захистили користувачів Firefox від збору відбитків третьою стороною. [37]

Але рівень захисту можна підвищити. Для цього потрібно зайти в налаштування браузера, шляхом введення в адресному рядку «about: config». Потім обрати і змінити наступні опції: [37]

- ✓ webgl.disabled – обрати «true». Це повинно допомогти запобігти відбитків Canvas. [5]
- ✓ geo.enabled – обрати «false». Завадить відстеженню геолокації. [5]
- ✓ privacy.firstparty.isolate – обрати «true». Це дозволить ізолювати файли cookie від сторонніх доменів. [37]
- ✓ privacy.resistFingerprinting – обрати «true». Ця опція дає базовий рівень захисту проти збору відбитків браузера. Але найбільш ефективна вона при виборі і інших опцій зі списку. [37]
- ✓ media.peerconnection.enabled – необов'язкова опція, але, якщо користувач працює з VPN (Virtual Private Network, віртуальна приватна мережа), її варто вибрати. Вона дає можливість запобігти витоку WebRTC і демонстрацію IP. [37]

▪ **Спеціалізовані розширення для браузера.** Розширення – делікатна тема, оскільки вони часом підвищують унікальність відбитку браузера. Використовувати їх чи ні – вибір користувача. [37]

Кілька розширень:

- ✓ Chameleon – модифікація значень User-gent. Можна встановити періодичність «раз в 10 хвилин», наприклад; [38]
- ✓ Canvasblocker – захист від збору цифрових відбитків з Canvas; [39]
- ✓ Canvas Defender – робить приблизно те ж, що і Canvasblocker; [40]
- ✓ User-Agent Switcher – робить приблизно те ж, що і Chameleon; [41]

Використовувати краще одне розширення, а не відразу всі.

▪ **Вимкнення Flash.** Flash розкриває багато даних про користувача, тому зараз розробники намагаються від нього відходити. Якщо користувач використовує найновіші версії веб-переглядача, йому нічого не потрібно робити. Усі вони за замовчуванням вимикають Flash. [37]

Якщо ж використовуються старіші версії, ось що потрібно зробити:

- ✓ Для Firefox перейти до «Додатки», потім «Плагіни» та обрати «Ніколи не активувати» для плагіна Shockwave Flash. [37]

✓ Для будь-якого браузера Chromium перейти у «Налаштування» та ввести «Flash» у рядку пошуку. Відбувається перехід до розділу «Налаштування сайту», і можна вибрати параметр, який зупиняє використання всіма сайтами Flash. [37]

- **Вимкнення JavaScript**. Як і Flash, JavaScript обмінюється великою кількістю конфіденційних даних про браузер та пристрій із веб-серверами. Як рішення проблеми – його вимкнення, але потрібно мати на увазі, що деякі вебсайти не працюватимуть належним чином, оскільки вони на нього покладаються. [37]

Brave за замовчуванням відключає JavaScript [5]. Для інших браузерів можна зробити наступне:

✓ Firefox – ввести «about: config» у рядку URL, натиснути «Accept the Risk and Continue», а потім ввести «javascript.enabled» у рядку пошуку. Двічі клацнути на ньому, щоб перемкнути «false». [37]

✓ Браузери Chromium – просто перейти до «Налаштування» та ввести «JavaScript» у рядку пошуку. Знову ж таки, відбувається перехід у «Налаштування сайту». Там можна перемкнути JavaScript на «Заблоковано». [37]

Крім того, можна використовувати NoScript, навіть налаштувати його, щоб дозволити JavaScript на сайтах, які дійсно потрібні. [37]

- **Використання Firefox із файлом user.js від ghacks**. Файл user.js – це файл конфігурації, який керує багатьма налаштуваннями Firefox. Файл від ghacks оптимізований для забезпечення конфіденційності та безпеки, тому не потрібно вручну обробляти about: config tweaks. [37]

Проте для використанням цього файлу необхідна наявність спеціалізованих знань та навичок. [37]

- **Браузер Brave.** Ще один браузер, який дає серйозний захист персональних даних. Браузер блокує різного роду трекери, використовує HTTPS і блокує скрипти. Крім того, Brave дає можливість блокувати велику частину інструментів для збору відбитків браузера (рис. 1.8). Проте тест на ентропію по-

казує понад 17,85 біт ідентифікаційної інформації, що є вище допустимого порогу. [37]

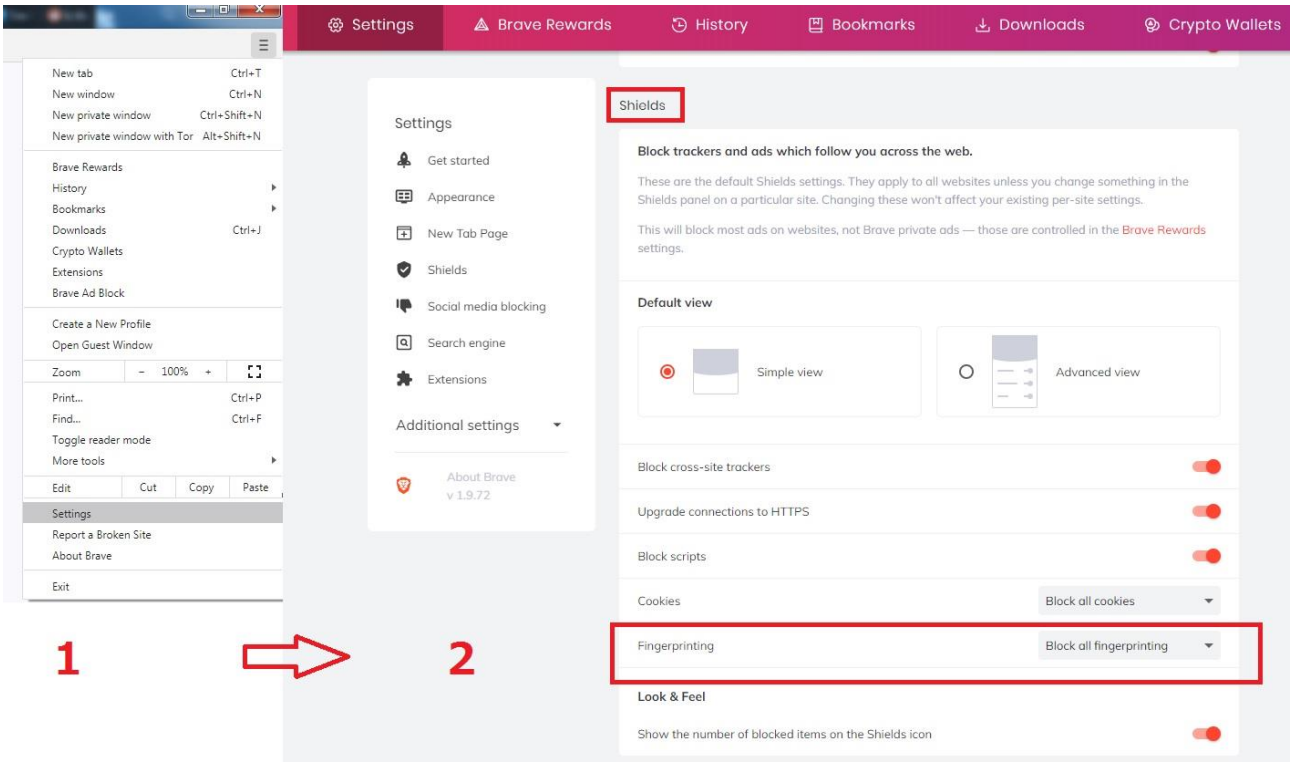


Рис. 1.8. Вікно налаштування захисту браузера Brave від відбитків браузера [37]

- **Tor браузер без Tor Network**. За замовчуванням браузер пропонує ряд інструментів для захисту персональних даних: [37]
  - ✓ HTTPS;
  - ✓ NoScript;
  - ✓ блокування WebGl;
  - ✓ блокування Canvas;
  - ✓ зміна версії ОС;
  - ✓ блокування інформації про часовий пояс і налаштування мови.

Але мережа Tor не вражає настільки, наскільки сам браузер. Ось чому:

– Працює мережа повільно. Все тому, що серверів – близько 6 тис., а користувачів – близько 2 млн. [37]

– Багато сайтів блокують трафік Tor – наприклад, Netflix. [37]

– Бувають витоки персональної інформації, одна з найсерйозніших трапилася в 2017 році. [37]

Загалом, є можливість використовувати Tor браузер без мережі Tor. Зробити це не так просто, але спосіб цілком доступний. Завдання – створити два файли, які відключать мережу Tor. [37]

Найкраще це робити в Notepad ++ (безкоштовний редактор вихідного коду [37]). Відкривши його, слід додати в першу вкладку такі рядки (рис.1.9):

Рис. 1.9. Вікно першої вкладки Notepad ++ [37]

Потім перейти в Edit – EOL Conversion, вибрати Unix (LF) і зберегти файл як autoconfig.js в директорію Tor Browser/defaults/pref. [37]

Потім відкрити нову вкладку і записати наступні рядки (рис.1.10):

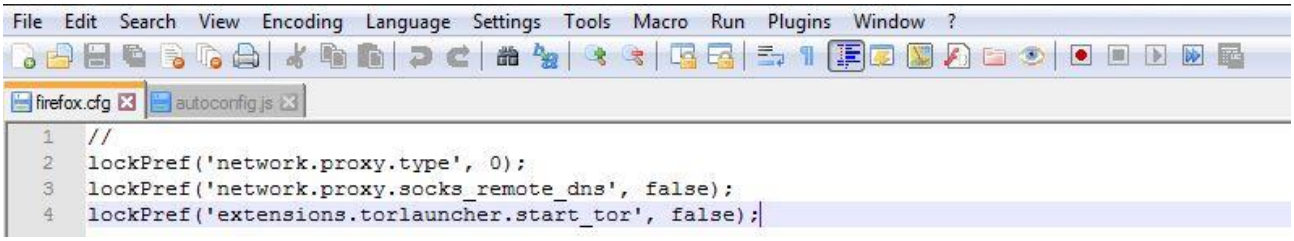


Рис. 1.10. Вікно другої вкладки Notepad ++ [37]

Назва файлу – firefox.cfg, його потрібно зберегти в Tor Browser/Browser.

Готово. Після запуску браузер покаже помилку, але на це можна не звертати уваги. Відключення мережі ніяк не вплине на відбиток браузера, але зробить користування і так доволі захищеним браузером Tor зручнішим, швидшим і безпечнішим. [37]

- **Використання віртуальної машини (ВМ)**. ВМ – це віртуальна ОС, яку користувач запускає поверх існуючої ОС. ВМ може зменшити ентро-

пію відбитків пальців, перешкоджаючи веб-сайтам отримувати відбитки пальців про ОС та апаратне забезпечення користувача. Вона діє як «посередник» між ОС та Інтернетом. [37]

«Обладнання» ВМ (графічний процесор, процесор) імітується, тому веб-сервери не збирають жодних даних, які можуть порушити конфіденційність користувача. [37]

Деякі з програм віртуалізації ОС:

- ✓ VirtualBox (Windows, Linux, macOS, Solaris); [42]
- ✓ VMware Workstation Player (Windows, Linux); [43]
- ✓ VMware Fusion (macOS); [44]
- ✓ Parallels Desktop (macOS); [45]
- ✓ Hyper-V (Windows). [46]

Недолік цього способу в тому, що користувачу потрібно буде отримати ліцензії для платних ОС, але завжди можна скористатися безкоштовною ОС, наприклад Ubuntu. [37]

- **Використання окремих пристроїв/браузерів**. Окремий пристрій (наприклад, ноутбук), використовувати для загального перегляду Інтернету (наприклад, для читання публікацій або розміщення дописів в соціальних мережах). [37]

Для особистих речей (електронної пошти, банківського рахунку, PayPal тощо) використовувати інший пристрій. В ідеалі слід запустити на ньому віртуальну машину та скористатися іншими порадами, про було згадано вище. [37]

Так само можна використовувати один браузер для публічної діяльності (соціальні медіа, розмови з друзями, пошукові запити, робота тощо), а інший браузер для приватної діяльності (Інтернет-банкінг, глибокі дослідження, криптовалюта тощо). [37]

Це ще не все – потрібно також використовувати окремі імена користувачів, адреси електронної пошти та паролі для свого громадського та приватного Інтернет-життя. В цілому це безпечніше, але занадто складно. [37]

Порівняння методів захисту від відбитка браузера, їх переваги і недоліки наведено в табл. 1.1. Враховуючи подану там інформацію, можна дійти висновку, що спеціалізовані розширення є найкращим способом захисту від відбитка браузера з найбільш прийнятним співвідношення плюсів і мінусів.

Таблиця 1.1

Порівняння існуючих методів боротьби з відбитком браузера

Метод	Переваги	Недоліки
Зміна налаштувань браузера	+ Найлегший і найшвидший спосіб; + можна застосувати до будь-якого браузера.	- Малоефективний і недовготривалий; - доволі складно для пересічного користувача.
Браузер Firefox з модифікованими налаштуваннями	+ Браузер дає можливість налаштувати захист від відбитків браузера на власний розсуд.	- Не найпопулярніший браузер; - суть методу полягає в налаштуваннях, які є доволі складними і незрозумілими для пересічного користувача.
Спеціалізовані розширення для браузера	+ Різноманітність вибору; + можливість налаштувати захист від відбитків браузера на власний розсуд; + в більшості випадків зручний і зрозумілий інтерфейс, майже не потрібно стикатися з налаштуваннями безпосередньо самого браузера; + можна застосувати до будь-якого браузера.	- Розширення часом підвищують унікальність відбитку браузера.
Браузер Brave	+ Дає можливість блокувати велику частину інструментів для збору відбитків браузера.	- Не найпопулярніший браузер; - тест на ентропію показує не найкращі результати.
Вимкнення Flash	+ Уникнення значної частки витоку інформації; + можна застосувати до будь-якого браузера.	- Суть методу полягає в налаштуваннях, які є доволі складними і незрозумілими для пересічного користувача; - нові браузери і так вимикають Flash.
Вимкнення JavaScript	+ Уникнення значної частки витоку інформації; + можна застосувати до будь-якого браузера.	- Суть методу полягає в налаштуваннях, які є доволі складними і незрозумілими для пересічного користувача; - деякі веб-сайти не працюватимуть належним чином, оскільки вони на нього покладаються.
Tor браузер без Tor Network	+ Браузер дає можливість налаштувати захист від відбитків браузера на власний розсуд.	- Суть методу полягає в налаштуваннях, які є доволі складними і незрозумілими для пересічного користувача і потребують додаткового ПЗ.
Використання Firefox із файлом user.js від ghacks	+ Не потрібно вручну змінювати налаштування; + постійні оновлення.	- Не найпопулярніший браузер; - складна і велика за обсягом документація.
Використання ВМ	+ Зменшує ентропію за рахунок передачі інформації про віртуальну, а не справжню ОС.	- Спосіб потребує додаткового ПЗ; - користувачу потрібно вчитися працювати з програмами віртуалізації ОС.
Використання окремих пристроїв/браузерів	+ Зрозумілий та досить дієвий спосіб.	- Спосіб потребує додаткового ПЗ у вигляді нових браузерів; - користувачу потрібно мати кілька пристроїв.

Серед наведених вище варіантів розширень всі захищають лише від одного параметру відбитка браузера (для Chameleon і User-Agent Switcher це User-Agent, а для Canvasblocker та Canvas Defender – відбиток Canvas) [38-41]. Отже, найкращий варіант – поєднати в одному розширенні захист від різних варіантів зчитування відбитка браузера, наприклад, захист від відбитка Canvas, WebGL, звукового відбитка, відстеження роздільної здатності екрану та апаратного відстеження.

1.3.3. Захист від відбитка браузера на законодавчому рівні.

Захист від відбитка браузера базується на:

- законах України:
  - «Про інформацію» від 02.10.1992 № 2657-XII; [47]
  - «Про захист інформації в інформаційно-телекомунікаційних системах» від 05.07.1994 № 80/94-ВР; [48]
  - «Про основні засади забезпечення кібербезпеки України» від 05.10.2017 № 2163-VIII; [8]
  - «Про захист персональних даних» від 01.06.2010 № 2297-VI. [49]
- постанові Кабінету Міністрів України:
  - «Про затвердження Правил забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах» від 29.03.2006 №373. [50]
- Нормативних документах в галузі технічного захисту інформації (НД ТЗІ) та державних стандартах України (ДСТУ):
  - ДСТУ 3396.1-96. Захист інформації. Технічний захист інформації. Порядок проведення робіт; [51]
  - НД ТЗІ 2.5-004-99. Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу; [52]
  - НД ТЗІ 2.5-005-99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу; [53]

- НД ТЗІ 2.5-010-03. Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу; [54]
- НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. [55]

1.4. Висновки до розділу 1

Інтернетом користуються мільярди людей [1], і їх безпека напряму залежить від ПЗ, що використовується для доступу до веб-сайтів та їх перегляду – веб-браузера [2]. Але користувачі часто нехтують налаштуваннями безпеки, в зв’язку з чим зростає загроза атак, які використовують вразливості веб-браузерів [3], такі атаки останнім часом стали популярним способом зловмисників компрометувати комп’ютерні системи [3].

Основною функцією веб-браузера є візуалізація HTML-коду [2], призначення веб-браузера – отримати вміст з Інтернету та відобразити його на пристрої користувача [2]. За роки існування браузери значно еволюціонували та змогли вдосконалити свої можливості і технології [16]. Найпопулярнішим браузером у світі на сьогоднішній день є Google Chrome, на другому місці Apple Safari, далі Edge, Mozilla Firefox, та Opera [17].

Серед розглянутих вразливостей браузера [6, 18-32] одним з найголовніших виявилось зчитування відбитків браузера, оскільки загроза порушення конфіденційності – основна причина, через яку користувач повинен бути уважний [3]. Відбиток браузера майже не змінюється з часом, що робить ідентифікацію практично стовідсотковою [33], він збирає величезний обсяг інформації [36]. Крім того налаштування захисту від інших вразливостей теж входить до відбитка браузера [33]. Зловмисник може використовувати дані про пристрій для застосування експлойтів чи «цифрового двійника», що несе значний збиток користувачу [34, 35].

На сьогоднішній день гарантовано дієвих методів і засобів захисту від технології зчитування відбитка браузера не розроблено [5], однак після аналізу заходів, застосування яких дозволяє радикально знизити унікальність веб-браузера [37], одним з кращих виявилося застосування спеціалізованих розширень, що дають різноманітність вибору; можливість налаштувати захист від відбитків браузера на власний розсуд; зручний і зрозумілий інтерфейс, через який майже не потрібно стикатися з налаштуваннями безпосередньо самого браузера; можливість застосування до будь-якого браузера. Проте вони часом підвищують унікальність відбитку браузера [37].

Всі розглянуті розширення захищають лише від одного параметру відбитка браузера (для Chameleon і User-Agent Switcher це User-Agent, а для Canvasblocker та Canvas Defender – відбиток Canvas) [38-41]. Отже, найкращий варіант – поєднати в одному розширенні захист від різних варіантів зчитування відбитка браузера, наприклад, захист від відбитка Canvas, WebGL, звукового відбитка, відстеження роздільної здатності екрану та апаратного відстеження.

РОЗДІЛ 2. РОЗРОБКА ВЛАСНОГО МОДУЛЮ ЗАХИСТУ ВІД ВІДБИТКА БРАУЗЕРА

Враховуючи результати аналізу методів захисту від відбитка браузера в розділі 1, можна дійти висновку, що найкращим рішенням є встановлення розширення. Проте представлені на сьогоднішній день розширення не дають комплексного захисту, тому в цьому напрямку і потрібно рухатися.

Дослідження [33] показало, що точно можуть бути ідентифіковані 74% настільних пристроїв, в той час як те ж саме можна сказати про всього лише 45% користувачів мобільних пристроїв (рис.2.1). Отже, комп'ютери більш вразливі до відслідковування.

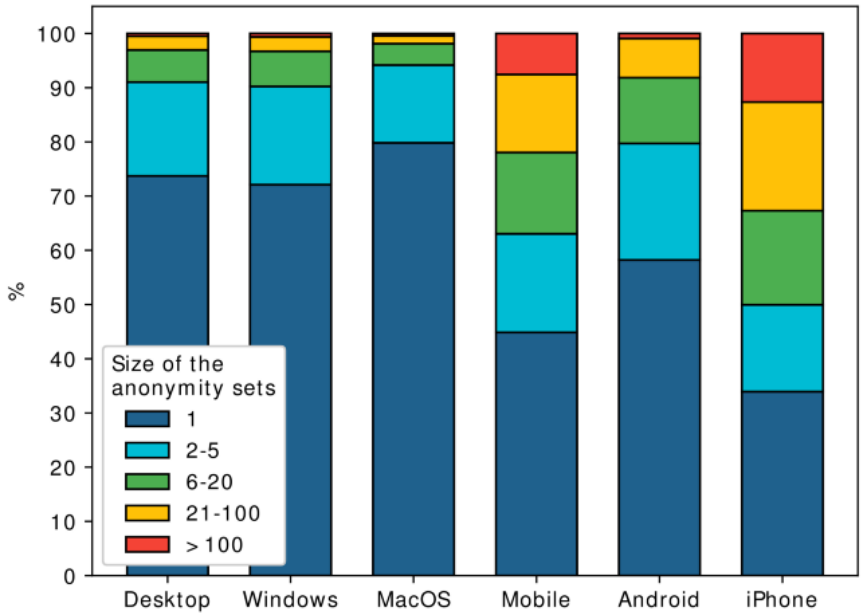


Рис. 2.1. Розміри анонімних груп для різних типів пристроїв [33]

Оскільки розроблений продукт буде вітчизняним, слід орієнтуватися на те, що в Україні станом на 2021 рік найпоширенішим веб-браузером є Google Chrome, на другому місці Opera, далі йде Mozilla Firefox, Apple Safari та Yandex Browser [56]. Таким чином розширення для Chrome є найбільш актуальним.

2.1. Браузер Google Chrome як об’єкт захисту від відбитка браузера, засоби вбудованого захисту в браузері Google Chrome

Політика конфіденційності Google [57] стверджує, що почати використовувати браузер Chrome можна без надання будь-якої особистої інформації. Проте мова йдеться лише про початок роботи. В деяких режимах браузер може збирати дані про користувача.

При роботі браузера в **основному режимі** на комп'ютері користувача зберігається така інформація: [57]

– Історія перегляду сайтів. Chrome зберігає URL відвіданих сторінок, файли кеша з текстом, картинками і іншим контентом зі сторінок, а також список IP-адрес деяких ресурсів, на які розміщені посилання на відвіданих сторінках (якщо включена попередня візуалізація).

– Персональні дані і паролі для автозаповнення форм та входу на сайти, де користувач уже бував раніше.

– Список дозволів, наданих веб-сайтам.

– Файли cookie або інші дані з відвіданих користувачем веб-сайтів.

– Дані, збережені доповненнями.

– Інформація про скачування з веб-сайтів.

Користувач може: [57]

- очищати історію відвідин сторінок;
- в будь-який момент видалити збережені дані, включаючи файли cookie та дані із сайтів;
- заборонити Chrome приймати файли cookie (але це призведе до того, що більшість сайтів, на яких потрібно входити в обліковий запис, не будуть працювати);
- переглядати паролі, збережені в Chrome;
- переглядати і змінювати збережені для автозаповнення дані.

Всі ці способи лише тимчасово допоможуть захиститися від відстеження. Крім того, особиста інформація, включаючи платіжні дані, карти і паролі, якщо користувач вирішив включити синхронізацію з Chrome відправляється в Google.

Як Chrome обробляє інформацію:

- **Інформація для власників сайтів**. Сайти, які користувач відкриває за допомогою Chrome, автоматично отримують IP-адресу і дані з файлів cookie. Якщо в Chrome буде виявлено, що користувач сервісу Google або партнерського сайту став жертвою мережевої атаки, то дані про це будуть спрямовані в Google або на той ресурс, де сталася атака, щоб визначити її серйозність. Власники сайтів, які співпрацюють з Google, отримують звіти про атаки, які були проведені на їх ресурсах. [57]
- **Попередня візуалізація.** Щоб швидше завантажувати веб-сторінки, Chrome може шукати IP-адреси посилань, розміщених на поточній сторінці, і створювати мережеві підключення. Запити від веб-сайтів виконуються завжди і не залежать від системи підказок Chrome. Якщо від браузера Chrome, веб-сайту або програми надійшов запит на попередню візуалізацію сторінки, вона буде зберігати і зчитувати файли cookie, ніби її вже відвідали. [57]
- **Місцезнаходження**. Щоб надати найбільш відповідну інформацію, деякі сайти можуть запитувати дані про місцезнаходження користувача. Проте слід зазначити, що Chrome надсилає такі дані лише з дозволу користувача. [57]
- **Оновлення**. Chrome іноді відправляє в Google запити, щоб перевірити оновлення, визначити статус з'єднання, синхронізувати налаштування часу і дізнатися кількість активних користувачів. [57]
- **Функції пошуку**. Якщо користувач увійшов в акаунт на сайті Google і використовує пошукову систему Google за замовчуванням, запити через універсальне вікно пошуку або вікно пошуку на сторінці швидкого доступу в Chrome будуть зберігатися в акаунті Google. [57]
- **Пошукові підказки**. Щоб користувач міг шукати інформацію швидше, Chrome пропонує варіанти запитів. Для цього він відправляє символи, які

користувач вводить в універсальне вікно пошуку навіть до того, як користувач натискає кнопку введення. Якщо користувач обрав Google в якості пошукової системи за замовчуванням, вона пропонує підказки на основі його історії пошуку та запитів інших людей. [57]

- **Автозаповнення, платежі і управління паролями**. Коли включено автозаповнення або управління паролями, Chrome надсилає в Google анонімну інформацію про веб-форми, які користувач відкриває або відправляє (в тому числі хеш URL веб-сторінки і дані про поля для введення). [57]
- **Мова**. Chrome запам'ятовує мову сайтів, які користувач відвідує найчастіше, і відправляє ці дані в Google. Якщо користувач включив синхронізацію Chrome, мовний профіль буде пов'язаний з його акаунтом Google. [57]
- **Статистика використання та звіти про помилки**. За замовчуванням ці дані відправляються в Google, щоб з їх допомогою компанія могла покращувати свої продукти. Статистика використання містить інформацію про налаштування, натискання кнопок і задіяні ресурси пам'яті. Як правило, в статистику не входять URL веб-сторінок і особиста інформація. Однак, якщо користувач ввімкнув функцію «Допомагати поліпшити перегляд сторінок і пошук» або «Відправляти URL відвіданих сторінок в Google», в статистиці будуть дані про те, які сторінки він відкривав і як їх використовував. Звіти про збої містять системну інформацію на момент збою, а також можуть включати URL вебсторінок і особисті дані. Google може передавати партнерам (наприклад, видавцям, рекламодавцям і розробникам) узагальнену інформацію, по якій можна встановити особу користувача. Слід зазначити, що користувач може в будь-який момент заборонити або знову дозволити Chrome відправляти в Google статистику і звіти. [57]

Ідентифікатори в Chrome

У Chrome використовуються різні унікальні і неунікальні ідентифікатори, необхідні для правильної роботи функцій:

- **Відстеження установок**. Кожна копія Chrome для Windows містить генерований випадковим чином номер, який відправляється в Google при першому запуску, а потім віддаляється при першому оновленні Chrome. [57]
- **Відстеження промоакцій**. Для контролю ефективності промоакцій Chrome генерує унікальний токен, який відправляється в Google при першому запуску і використанні браузера. [57]
- **Тестування**. Іноді компанія організовує закриті тести нових функцій. При першому запуску браузеру присвоюється випадковий ідентифікатор, який потрібен для вибору фокус-груп. Тестування може бути обмежене країною (визначається за IP-адресою), ОС, версією Chrome та іншими параметрами. [57]

Режим інкогніто і гостьовий режим

Користувач може скористатися режимом інкогніто або гостьовим режимом, щоб обмежити обсяг інформації, який Chrome зберігає у на комп'ютері. У цих режимах деяка інформація не зберігається, наприклад: [57]

- основна інформація про історію перегляду сайтів, включаючи URL, кеш тексту сторінок і IP-адреси, пов'язані з відвідуваними веб-сайтами;
- зменшені зображення відвідуваних сайтів;
- записи про скачування файлів.

Порядок обробки інформації в режимі інкогніто і гостьовому режимі:

- ✓ **Файли cookie.** Chrome не надає сайтам доступ до файлів cookie при роботі в цих режимах. Сайти можуть зберігати в системі нові файли cookie, але всі вони будуть видалені, коли користувач закриє вікно браузера. [57]
- ✓ **Зміни конфігурації браузера**. Коли користувач вносить зміни в конфігурацію браузера, наприклад створює закладку для веб-сторінки або змінюєте налаштування, ця інформація не зберігається. [57]
- ✓ **Дозволи**. Дозволи, які користувач надає в режимі інкогніто, не зберігаються в існуючому профілі. [57]
- ✓ **Інформація з профілю**. У режимі інкогніто у користувача є доступ до інформації з існуючого профілю, наприклад до підказок на базі історії відві-

дування сторінок та збережених паролів. У гостьовому режимі дані профілів не використовуються. [57]

Безпечний перегляд веб-сторінок

Google Chrome підтримує функцію Безпечного перегляду Google. При безпечному перегляді браузер отримує від серверів Google інформацію про підозрілі веб-сайти. [57]

Порядок роботи функції Безпечного перегляду

Браузер періодично звертається до серверів Google для завантаження постійно оновлюваного списку сайтів, помічених у фішингу та поширенні шкідливого ПЗ. Поточна копія списку зберігається у системі користувача локально. При цьому в Google не надходять ні відомості про акаунт, ні інші ідентифікаційні дані. Передається IP-адреса і файли cookie. [57]

Кожен відвіданий користувачем сайт звіряється з завантаженим списком. При виявленні відповідностей браузер відправляє в Google частковий хеш копії URL, щоб отримати додаткову інформацію. [57]

Ряд функцій Безпечного перегляду працює тільки в Chrome:

- Якщо користувач включив режим поліпшеного захисту за допомогою Безпечного перегляду, в Chrome використовуються додаткові засоби захисту. При цьому в Google відправляється більше даних. [57]
- Якщо користувач включив Безпечний перегляд, а також параметр «Допомагати поліпшити перегляд сторінок і пошук/Відправляти URL відвіданих сторінок в Google», Chrome надсилає в Google повний URL кожного відкритого користувачем сайту. [57]
- Chrome використовує технологію Безпечного перегляду, щоб періодично сканувати комп'ютер користувача, виявляючи небажані програми. [57]
- Щоб допомогти компанії вдосконалити режим Безпечного перегляду, користувач може налаштувати відправку додаткових даних. Вони будуть передаватися при переході на підозрілий сайт або при виявленні небажаного ПЗ на комп'ютері. [57]

У Chrome можна користуватися **розширеннями, темами, сервісами та іншими додатками**, включаючи встановлені або інтегровані. Додатки, розроблені і надані компанією Google, можуть відправляти дані на її сервери і контролюються Політикою конфіденційності Google, якщо не вказано інше. Сторонні додатки контролюються їх розробниками, у яких може діяти інша політика конфіденційності. [57]

Враховуючи, скільки інформації від користувача отримує браузер Google Chrome і який захист при цьому пропонує, при користуванні ним просто необхідний додатковий захист від відбитка браузера.

2.2. Авторський варіант захисту від відбитка браузера

Отже, вирішено розробити розширення для браузера Google Chrome для настільних пристроїв, яке буде поєднувати захист від різних параметрів відбитка браузера.

Розширення браузера – це невеликий програмний модуль для налаштування веб-браузера та розширення його функціональних можливостей. Розширення не можна плутати з плагінами та доповненнями. Синтаксис для розширень може відрізнятися для різних браузерів. Браузер Google Chrome дозволяє встановлювати різноманітні розширення, включаючи модифікації користувальницького інтерфейсу, блокування реклами та керування файлами cookie.

Для реалізації обрані такі засоби:

- середовище розробки програмного додатку WebStorm;
- фреймворк (програмний каркас) AngularJS;
- мови програмування:
  - CSS;
  - HTML;
  - JavaScript.

Для тестування обрано сервіс Panopticlick [58].

2.2.1. Середовище розробки програмного додатку WebStorm.

WebStorm – це інтегроване середовище розробки для кодування в JavaScript та пов'язаних з ним технологіях, включаючи TypeScript, React, Vue, Angular, Node.js, HTML та CSS. [59]

WebStorm забезпечує підсвічування і автодоповнення коду, його аналіз під час редагування і швидку навігацію. Він має потужні інструменти налагодження та інтеграції з системами управління версіями, розуміє структуру проекту і код, відстежує помилки за допомогою різних систем і пропонує їх рішення. Також в WebStorm є вбудовані інструменти для тестування. [59]

Інтерфейс користувача WebStorm показано на рис. 2.2. Вікно WebStorm складається з редактора, де користувач читає, створює та модифікує свій код, меню, панелі інструментів, панелі навігації та рядка стану.

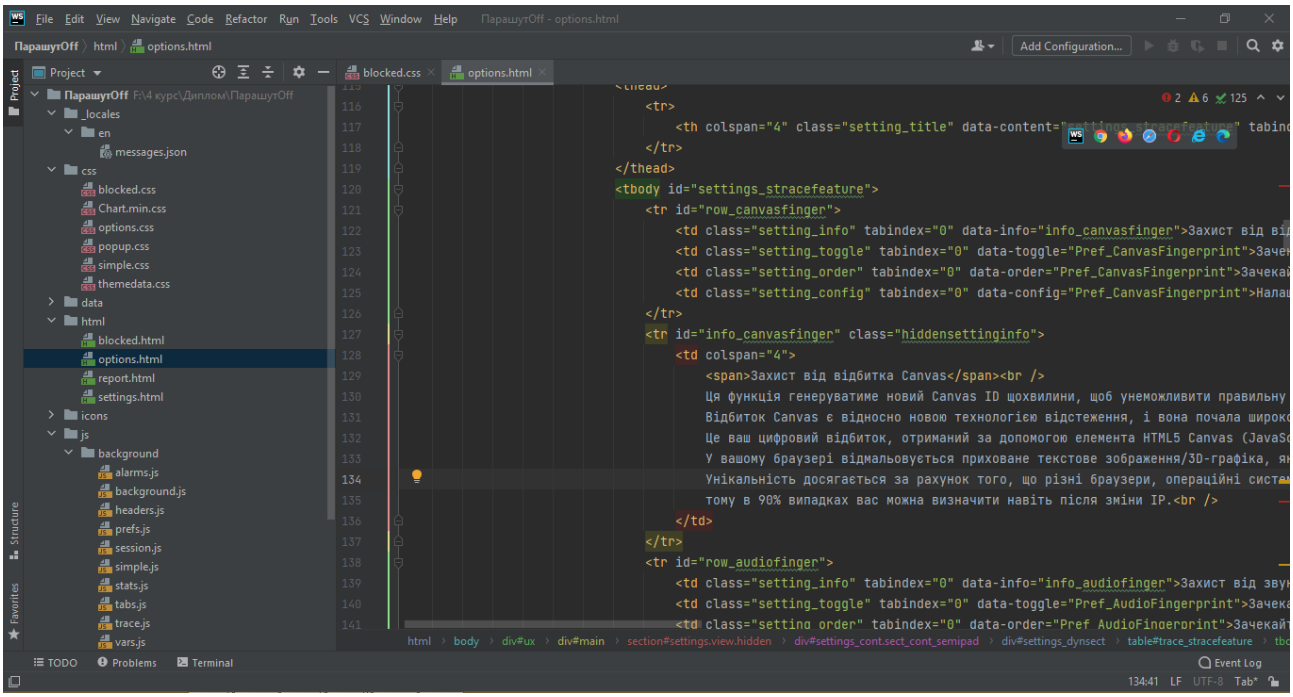


Рис. 2.2. Інтерфейс WebStorm

2.2.2. Фреймворк AngularJS.

Фреймворк – програмна платформа, яка визначає структуру програмної системи; ПЗ, що полегшує розробку і об'єднання різних компонентів великого програмного проекту. [60]

AngularJS – вільний JavaScript-фреймворк з відкритим вихідним кодом. Призначений для розробки односторінкових додатків. Його мета – розширення браузерних додатків на основі MVC-шаблону (Model-view-controller, Модель-вигляд-контролер), а також спрощення тестування і розробки. [60]

Фреймворк працює з HTML, що містить додаткові атрибути, які описуються директивами, і пов'язує введення або виведення області сторінки з моделлю, яка представляє собою звичайні змінні JavaScript. Значення цих змінних задаються вручну або витягуються з статичних або динамічних JSON-даних (JavaScript Object Notation, запис об'єктів JavaScript). [60]

Фреймворк адаптує і розширює традиційний HTML, щоб забезпечити двосторонню прив'язку даних для динамічного контенту, що дозволяє автоматично синхронізувати модель і уявлення. В результаті AngularJS зменшує роль DOM-маніпуляцій (Document Object Model, об'єктна модель документа) і покращує тестування. [60]

2.2.3. Мови розробки.

Для розробки розширення використовувались три мови програмування: CSS, HTML та. JavaScript. Всі вони так чи інакше взаємодіють між собою.

HTML – мова розмітки веб-документів та веб-сторінок. HTML інтерпретується браузерами, отриманий в результаті цього форматований текст відображається на екрані монітора комп'ютера. В Інтернеті HTML-сторінки передаються браузерам від сервера по протоколах HTTP або HTTPS у вигляді простого тексту або з використанням шифрування. [12]

В HTML вбудовується програмний код на мові програмування JavaScript для управління поведінкою і змістом веб-сторінок, а також CSS, щоб описати зовнішній вигляд і макет сторінки. [12]

CSS – спеціальна мова стилю сторінок, використовується для опису зовнішнього вигляду документа чи веб-сторінки, написаних мовою розмітки даних (в даному випадку HTML). В розширеннях CSS застосовується, щоб визначити шрифт, колір, розташування об’єктів на сторінках розширення. [12]

JavaScript в розширеннях використовується як мова сценаріїв для додання інтерактивності веб-сторінок, а також щоб налаштувати взаємодію з користувачем. [25]

2.2.4. Сервіс Panopticlick.

Сервіс Panopticlick від Electronic Frontier Foundation [58] використовується для визначення унікальності браузера. Він визначає, які дані про користувача зчитує відбиток браузера, а також кількість бітів ідентифікаційної інформації. Якщо їх сума перевищує 17,85 біт, то користувача можна визначити з майже стовідсотковою ймовірністю.

Початкова сторінка сервісу показана на рис. 2.3.

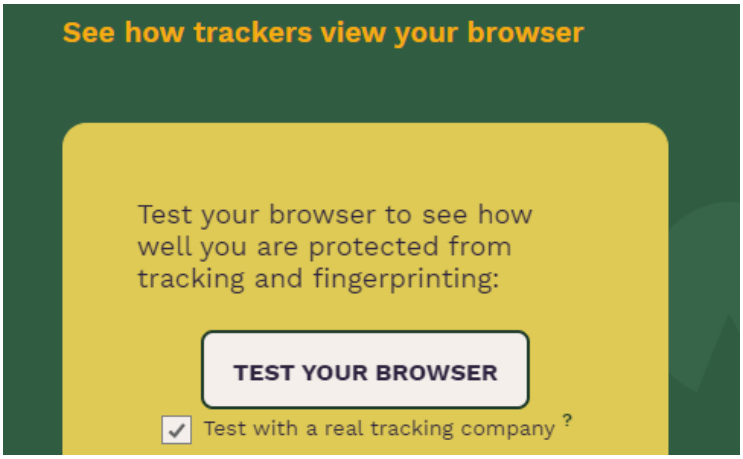


Рис. 2.3. Вікно початкової сторінки сервісу Panopticlick

2.2.5. Створення розширення «ПарашутOff» для браузера

Спочатку потрібно визначити місце розширення в схемі збору відбитків браузера. На рис. 2.4 видно, що спочатку диспетчер завдань на стороні сервера надсилає різні завдання візуалізації, такі як малювання кривих та ліній, на сторону клієнта. Завдання візуалізації включають інформацію про ОС та апаратне забезпечення, наприклад, роздільну здатність екрана та часовий пояс. Потім браузер на стороні клієнта виконує ці завдання і показує відповідні результати, наприклад, зображення та звукові хвилі. Потім ці результати, особливо зображення, перетворюються в хеші, щоб їх було зручно відправити на сервер. Тим часом браузер також збирає специфічну інформацію для складання відбитків

браузера. Далі сервер почне складати відбитки браузера за допомогою маски, яка являє собою список одиниць та нулів – виконується операція «AND» між списком хешів та маскою, а потім генерується ще один хеш як відбиток браузера. Розширення ж буде перехоплювати інформацію, що збирається на стороні клієнта, і змінювати її так, щоб якомога більше знизити унікальність, а вже потім передавати серверу для формування відбитка браузера.

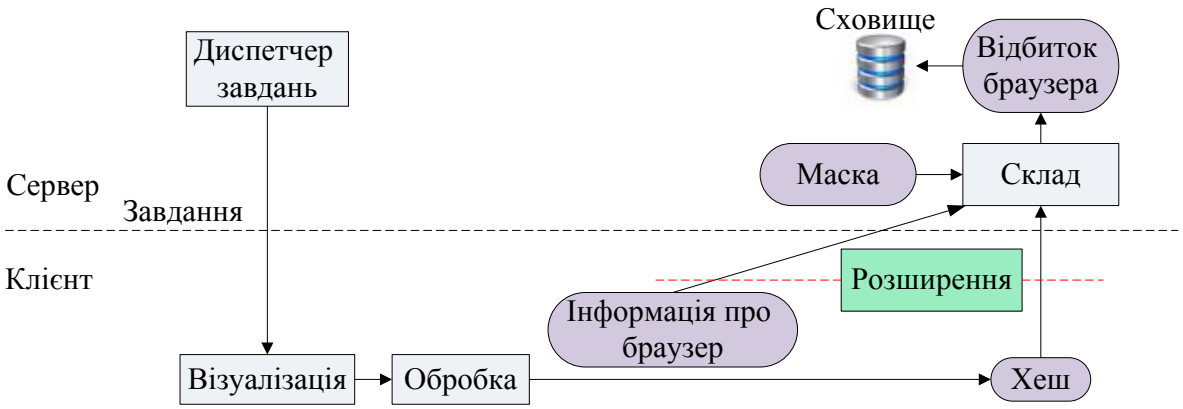


Рис. 2.4. Місце розширення в схемі збору відбитків браузера

На рис. 2.5 зображена схема роботи розширення.

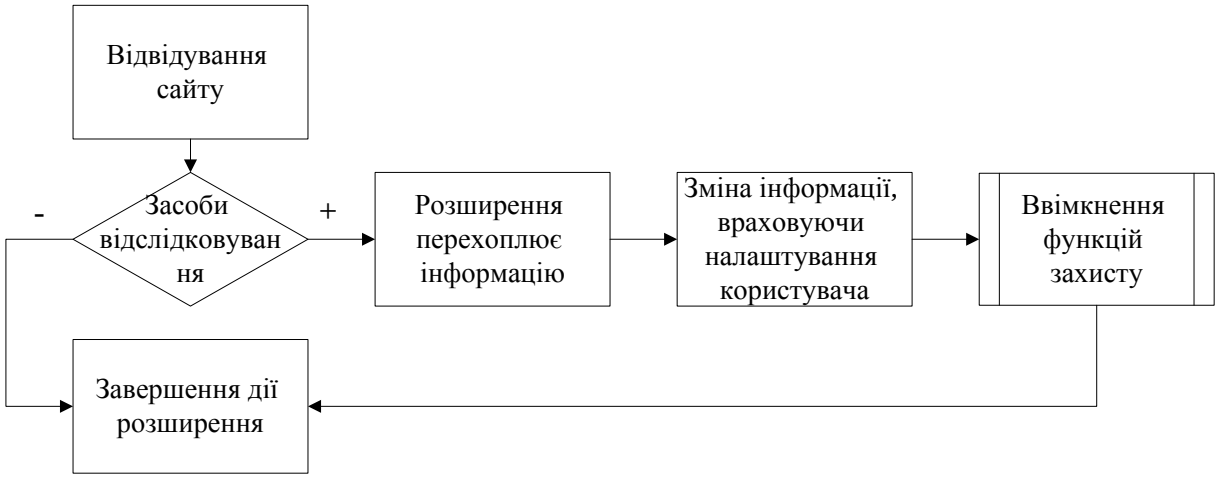


Рис. 2.5. Схема роботи розширення

Будь-яке Chrome-розширення починається з маніфесту – головного файлу в Chrome-розширенні, оскільки в ньому зберігається інформація про потрібні доступи, файли, які підключаються до сторінки, налаштування безпеки і багато іншого.

На рис. 2.6 показано фрагмент коду маніфесту розробленого розширення, де вказано такі дані:

- manifest_version – версія маніфест-файлу;
- name – назва розширення;
- author – автор розширення;
- description – опис розширення;
- default_locale – файл мови за замовчуванням;
- minimum_chrome_version – мінімальна версія Google Chrome, що підтримує розширення
- version – версія розширення;
- browser_action – налаштування відповідної кнопки на панелі інструментів;
- default_icons – списки іконок за стандартними розмірами;
- default_title – заголовок розширення за замовчуванням.

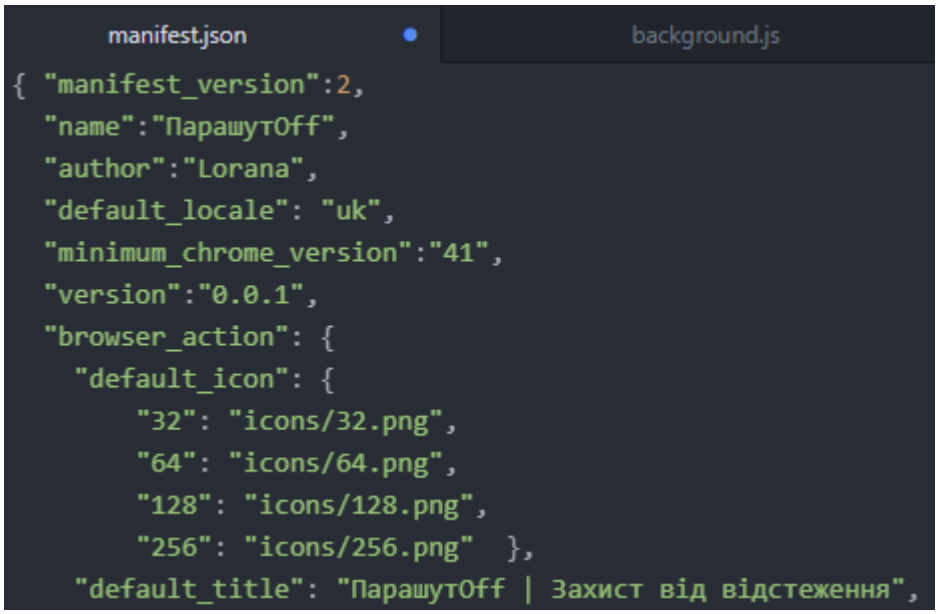


Рис. 2.6. Фрагмент коду маніфесту розширення ПарашутOff

Наступний крок – вибір іконок для розширення. Зображення повинно бути формату png (Portable Network Graphics, портативна мережева графіка).

На рис. 2.7 показано іконки розширення ПарашутOff.

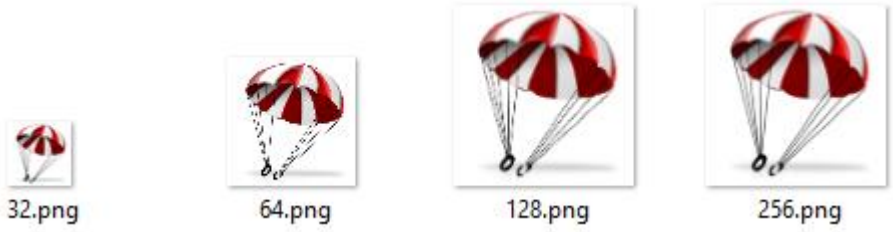


Рис. 2.7. Іконки розширення ПарашутOff

Після того, як розширення описано у файлі manifest.json, і підібрано іконки, можна переходити до наступного етапу, а саме до розмітки. На рис. 2.8 показано приклад використання мови HTML для розмітки. Спочатку вказано тег <html>, який повідомляє браузеру про те, де починається HTML-код, і де він закінчується. Цей тег є кореневим елементом і використовується як контейнер для всіх інших HTML-елементів. Далі йде тег <head>, що містить в собі опис веб-сторінки і є контейнером для всіх елементів-заголовків. Тег <title> не відображається напряму на веб-сторінці, але повинен бути обв’язково присутній в коді, щоб відображатися в назві вкладки. Тег <link> посилається на зовнішній ресурс, в даному випадку на файли CSS.

```
<html>
  <head>
    <title>Заблоковано ПарашутOff</title>
    <link rel="stylesheet" media="all" type="text/css" href="../css/blocked.css" />
    <link rel="stylesheet" media="all" type="text/css" href="../css/themedata.css" />
  </head>
```

Рис. 2.8. Фрагмент коду, написаний мовою HTML

Повний список HTML-файлів показано на рис. 2.9.

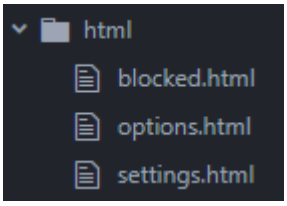


Рис. 2.9. Список HTML-файлів розширення ПарашутOff

Щоб розширення мало зручний і зрозумілий інтерфейс, потрібно додати стилі, написані мовою CSS.

До прикладу, в фрагментові коду на рис. 2.10 заголовок другого рівня h2 буде відображатися білим кольором на градієнтному тлі (чорний перетікає в зелений, потім знову в чорний), зі збільшеним розміром.

```
h2 {
  font-size:2.4em;
  color:#fff;
  background:linear-gradient(to top, #000000, #018e84, #000000);}
```

Рис. 2.10. Фрагмент коду, написаний мовою CSS

Як це виглядає в інтерфейсі користувача показано на рис. 2.11.

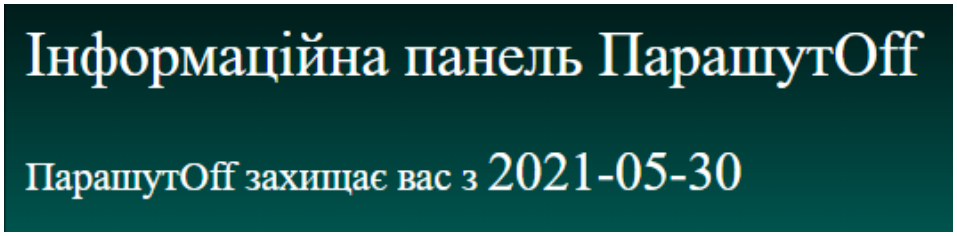


Рис. 2.11. Вигляд заголовку другого рівня в інтерфейсі користувача

Повний список CSS-файлів показано на рис. 2.12.

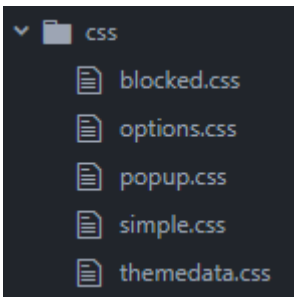


Рис. 2.12. Список CSS-файлів розширення ПарашутOff

Далі можна забезпечити функціонал сторінок за допомогою JavaScript. На рис. 2.13 показано функцію відкриття діалогу, де розширення запитує у користувача значення RGBA (red green blue alpha, червоний зелений синій альфа) для того, щоб змінити ці значення у відбитку Canvas.

```
CanvasRGBA:{
  OpenDialog:function(){
    let opts = {
      "type":"checkbox",
      "id":"trcanv_custrgba"
    };
    let rgba = Opts.Config.CurrentSel["customRGBA"]["rgba"];
    if (Opts.Config.CurrentSel["customRGBA"].enabled === true){
      opts["checked"] = "checked";
    }
```

Рис. 2.13. Фрагмент коду з підміною значень RGBA відбитка Canvas

Як це виглядає в інтерфейсі користувача показано на рис. 2.14.

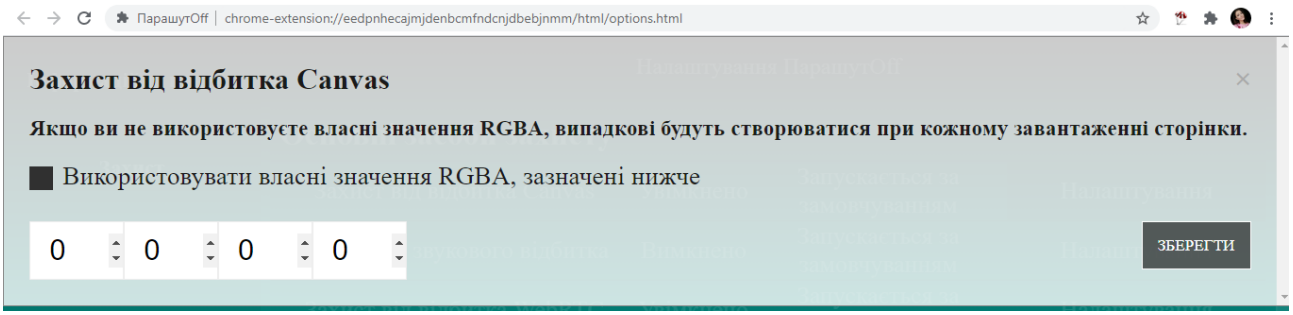


Рис. 2.14. Підміна значень RGBA відбитка Canvas в інтерфейсі користувача

На рис. 2.15 показано функцію вибору налаштувань рандомайзера User-Agent, де користувач може самостійно обрати ОС, яку буде зчитувати відбиток браузера.

```
ChooseUserAgent:function(){
  if (Prefs.Current.Pref_UserAgent.uaCust.enabled === true && Prefs.Current.Pref_UserAgent.uaCust.customUAs.length > 0){
    return rA(Prefs.Current.Pref_UserAgent.uaCust.customUAs); }
  let uaOSPool = [];
  if (Prefs.Current.Pref_UserAgent.uaOSConfig.AllowLinux.enabled === true){
    uaOSPool = uaOSPool.concat(Object.values(Vars.uaSettings.os.linux));  }
  if (Prefs.Current.Pref_UserAgent.uaOSConfig.AllowMac.enabled === true){
    uaOSPool = uaOSPool.concat(Object.values(Vars.uaSettings.os.macos));  }
  if (Prefs.Current.Pref_UserAgent.uaOSConfig.AllowWindows.enabled === true){
    uaOSPool = uaOSPool.concat(Object.values(Vars.uaSettings.os.windows));  }
  let uaWBPool = [];
  if (Prefs.Current.Pref_UserAgent.uaWBConfig.AllowChrome.enabled === true){
    uaWBPool = uaWBPool.concat(Object.values(Vars.uaSettings.wb.chrome)); }
  if (Prefs.Current.Pref_UserAgent.uaWBConfig.AllowFirefox.enabled === true){
    uaWBPool = uaWBPool.concat(Object.values(Vars.uaSettings.wb.firefox)); }
```

Рис. 2.15. Фрагмент коду з вибором налаштувань рандомайзера User-Agent

Як це виглядає в інтерфейсі користувача показано на рис. 2.16.

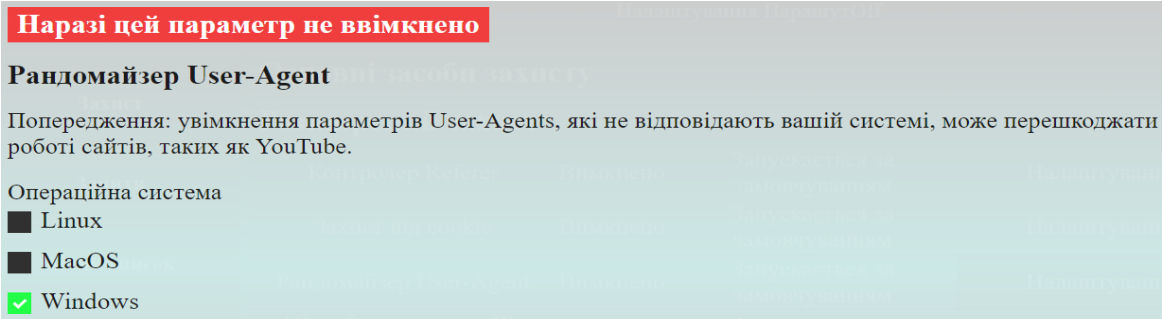


Рис. 2.16. Вибір налаштувань User-Agent в інтерфейсі користувача

На рис. 2.17 показано функцію підміни IP-адреси, де розширення використовує випадкову IP-адресу, а у випадку помилки – IP-адресу 128.128.128.128.

```
if (Prefs.Current.Pref_IPSpoof.traceIP.enabled === true && chrome.alarms){
  Trace.i.RefreshFakeUserIP();
  chrome.alarms.create("UserFakeIPRefresh", {periodInMinutes: Vars.FakeIPInterval});
  if (Trace.DEBUG) console.log("[pipsd]-> Використання випадкового IP");
} else {
  if (!onlyStart){
    Trace.i.StopIPRefresh();
    Headers.IPSpoof.Stop(); }
  if (typeof Prefs.Current.Pref_IPSpoof.traceIP.user_set !== "string" || Prefs.Current.Pref_IPSpoof.traceIP.user_set.length < 7){
    Prefs.Set("Pref_IPSpoof.traceIP.user_set","128.128.128.128");
    Trace.i.CurrentFakeIP = "128.128.128.128";
```

Рис. 2.17. Фрагмент коду з підміною IP-адреси

На рис. 2.18 показано функцію перехоплення і читання заголовків cookie з метою їх обробки. Код обробки взятий із фреймворка AngularJS, де було вказано, які саме файли cookie потрібно перехоплювати.

```
var CookieParser = function(string){
  this.cookies = string || "";
  this.inital = string;
  this.parsed = {};

  this.decodeuri = function(s){
    if (!s) return "";
    return s.replace(/(%[0-9A-Z]{2})+/g, decodeURIComponent);
  };

  this.parse = function(){
    var r = {};
    var decoded = this.cookies ? this.cookies.split('; ') : [this.cookies];
    var i = 0, l = decoded.length;
```

Рис. 2.18. Фрагмент коду перехоплення і читання заголовків cookie з метою їх обробки

Для того, щоб в майбутньому можна було перемикати мову розширення, доречно створити окремий файл для повідомлень українською (рис. 2.19).

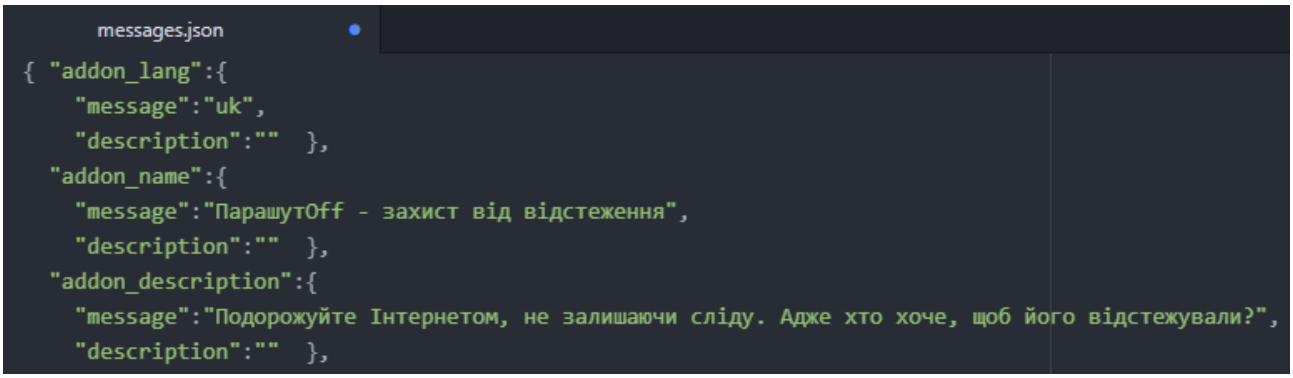


Рис. 2.19. Файл з повідомленнями українською мовою

В ПарашутOff вбудована функція блокування шкідливих веб-сайтів. Список таких сайтів взято із вільного доступу [38] і показано на рис. 2.20.

Рис. 2.20. Фрагмент коду зі списком шкідливих веб-сайтів

2.2.6. Рекомендації щодо використання розширення.

Дане розширення рекомендовано застосовувати як на підприємствах, так і для захисту пересічного користувача. ПарашутOff дозволить убезпечити комп'ютерну систему від цільової реклами та попередити зловмисницькі дії щодо витоку конфіденційної інформації.

Порядок встановлення і користування:

1. **Розпакувати розширення.**

Перед початком роботи слід розпакувати архів з розширенням, натиснувши на нього правою кнопкою миші і обравши потрібний параметр зі списку: вилучити файли або вилучити в поточну папку (рис. 2.21).

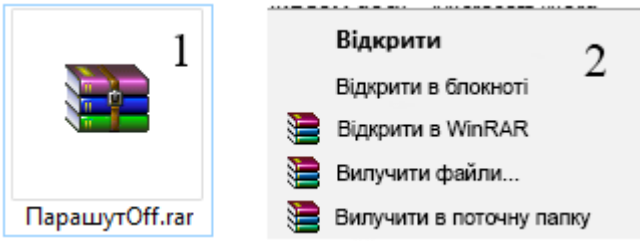


Рис. 2.21. Розпакування розширення

Якщо користувач перемістить цю папку, налаштування розширення будуть скинуті, тому обов'язково потрібно створити резервну копію, перш ніж це зробити.

2. Перейти на панель «Розширення» в браузері Chrome.

Для цього є кілька шляхів:

- набрати chrome://extensions в пошуковому рядку (рис. 2.22, *а*);
- натиснути на три точки в правому верхньому куті вікна браузера та перейти за шляхом «Додаткові інструменти» – «Розширення» (рис. 2.22, *б*);
- якщо користувач уже встановлював інші розширення, то можна натиснути на значок розширення поруч з пошуковим рядком і перейти в «Управління розширеннями» (рис. 2.22, *в*).

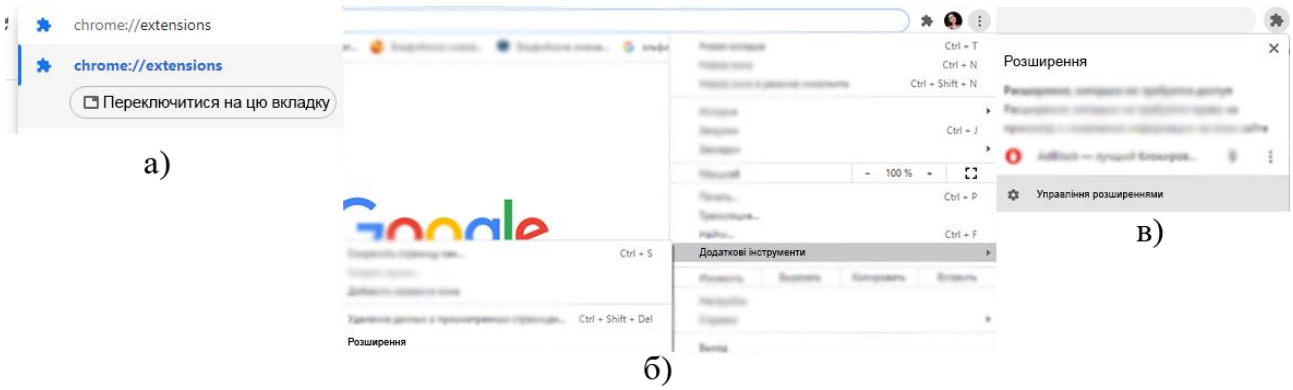


Рис. 2.22. Порядок переходу на панель «Розширення» в браузері Chrome: *а)* перший варіант; *б)* другий; *в)* третій.

3. Увімкнути режим розробника.

Це можна зробити на панелі «Розширення в правому верхньому куті (рис. 2.23). Якщо режим розробника уже ввімкнено, крок можна пропустити.

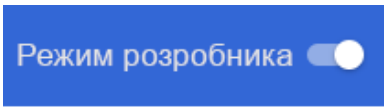


Рис. 2.23. Увімкнення режиму розробника

4. Завантажити розпаковане розширення.

Після увімнення режиму розробника з’явиться панель з кнопками «Завантажити розпаковане розширення», «Упакувати розширення» і «Обновити». Слід натиснути першу (рис. 2.24).

Рис. 2.24. Завантаження розпакованого розширення

Відкриється вікно «Оберіть каталог розширення», де потрібно обрати розпаковану папку «ПарашутOff» (рис. 2.25).

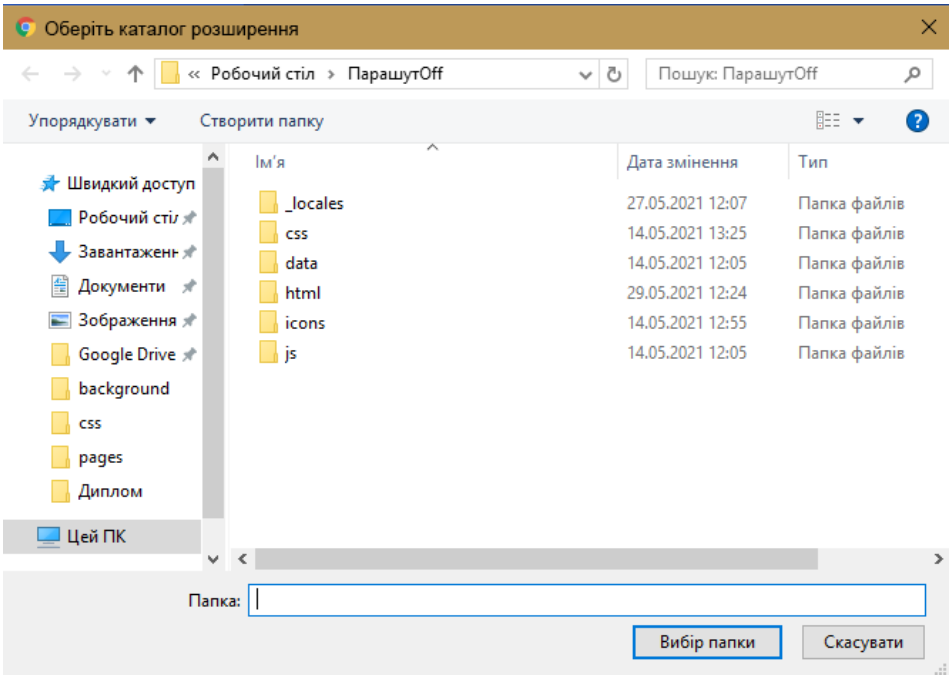


Рис. 2.25. Вибір каталогу розширення

Після цього розширення ПарашутOff з’явиться серед інших (рис. 2.26).

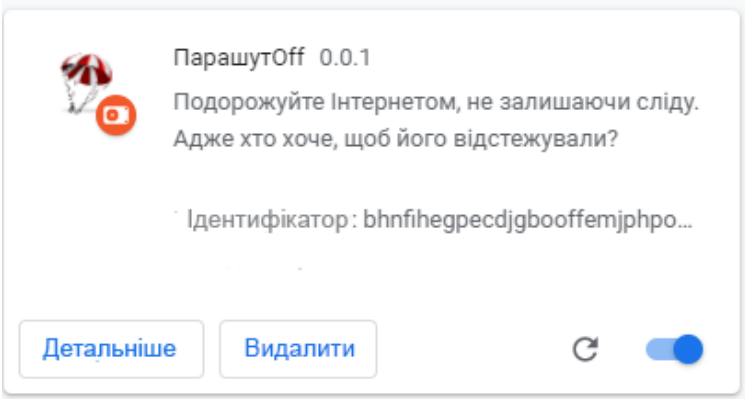


Рис. 2.26. ПарашутOff на панелі «Розширення»

5. Ознайомитися з установленим розширенням.

Розширення ПарашутOff розроблене так, щоб бути максимально зручним і зрозумілим. При першому встановленні розширення відразу перекидає користувача на сторінку налаштувань, де поверх неї висвічується привітання з поясненнями (рис. 2.27).

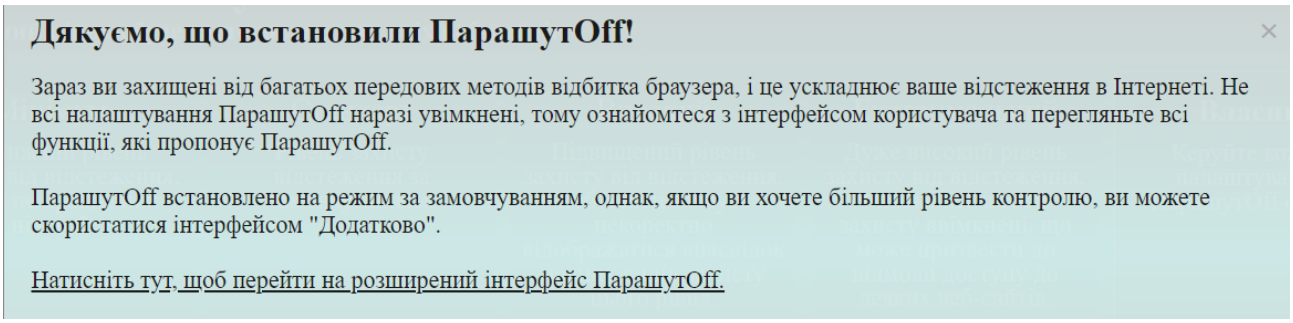


Рис. 2.27. Привітання розширення ПарашутOff

За замовчуванням розширення вмикає захист від відбитка Canvas, відбитка WebRTC, від апаратного відстеження і відстеження роздільної здатності екрану. Всі ці налаштування ніяк не впливають на відображення сторінок. Захист можна посилити, перейшовши до розширеного інтерфейсу, але посилення заходів безпеки може вплинути на відображення сторінок.

Розширення підтримує п’ять режимів захисту від відбитка браузера: мінімум, стандарт, високий, екстремальний і власний.

Режими розширення показано на рис. 2.28. Вони розміщені на сторінці налаштувань.

Рис. 2.28. Режими розширення ПарашутOff

Перейшовши на розширений інтерфейс, можна побачити інформаційну панель (рис. 2.29), де розташована інформація, коли розширення встановлене, а також можливість призупинити розширення (після перезавантаження браузера воно знову ввімкнеться).

Рис. 2.29. Інформаційна панель розширення ПарашутOff

Панель «Захист» (рис. 2.30) містить всі налаштування захисту від відбитка браузера. На панелі знаходяться дві вкладки: «Основні засоби захисту» і «Розширений захист» (більший захист – більше помилок сайтів). При натисканні на захист можна відкрити його опис. Кожен захист можна ввімкнути/вимкнути, а також додатково налаштувати, щоб зробити відбиток якомога менш унікальним.

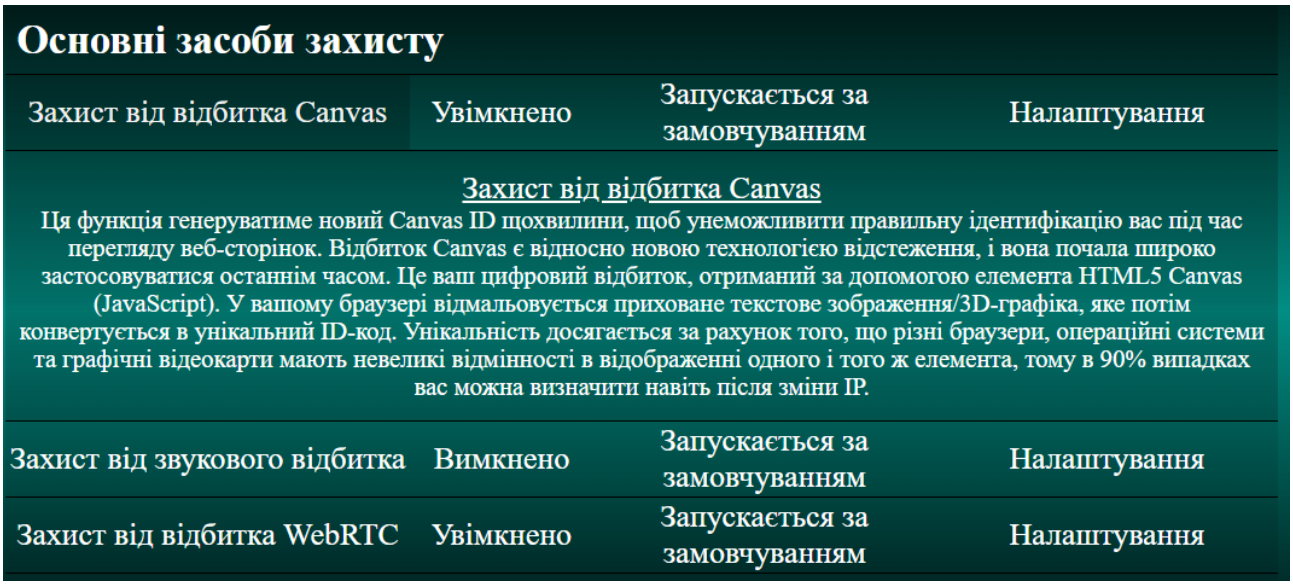


Рис. 2.30. Панель «Захист» розширення ПарашутOff

Панель «Запити» (рис. 2.31) містить локальний список блокування шкідливих сайтів, захист від шкідливих TLD (top-level domain, домен верхнього рівня) і засіб від відстеження URL-адрес. Кожен із засобів можна ввімкнути/вимкнути.

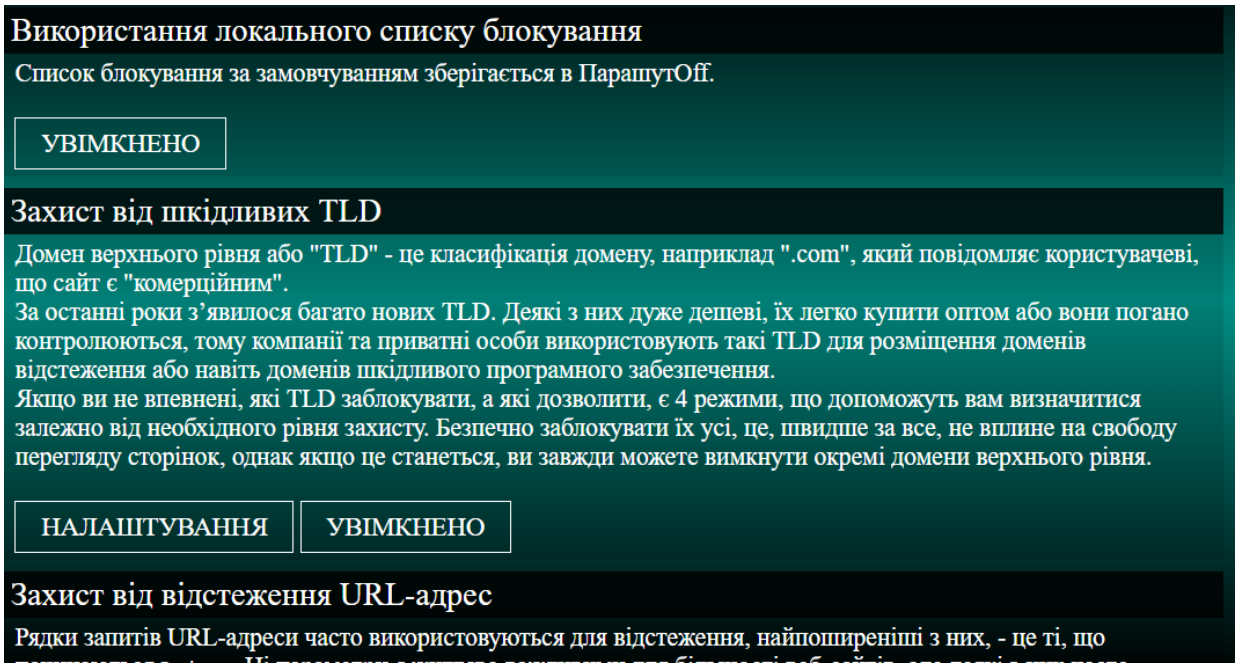


Рис. 2.31. Панель «Запити» розширення ПарашутOff

Будь-який заблокований сайт можна занести до білого списку на панелі «Білий список».

Панель «Параметри» (рис. 2.32) містить налаштування безпосередньо самого розширення: можна скинути налаштування до налаштувань за замовчуванням, увімкнути сповіщення браузера, а також гарячі клавіші.

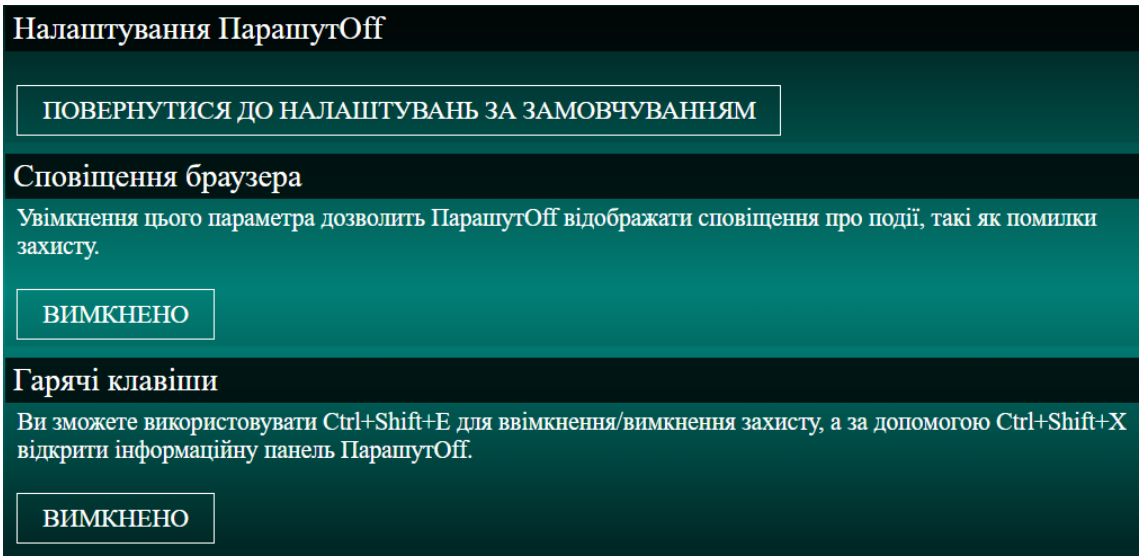


Рис. 2.32. Панель «Параметри» розширення ПарашутOff

2.2.7. Тестування створеного розширення на предмет ефективності захисту від відбитка браузера.

Початковий тест без будь-якого захисту від відбитка браузера показав результати, представлені на рис. 2.33.

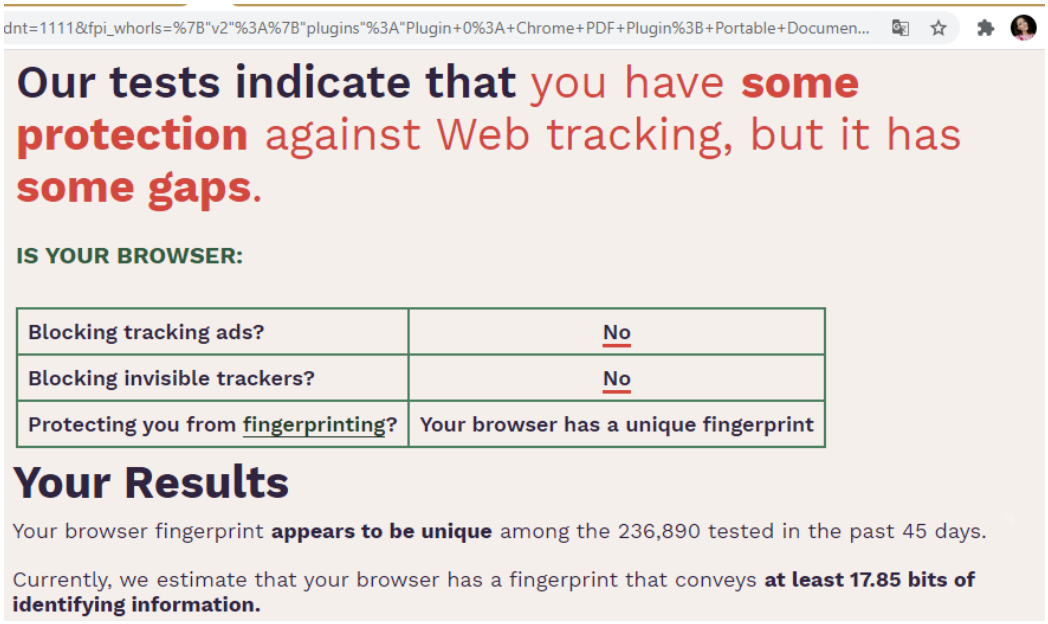


Рис. 2.33. Тестування до ввімкнення захисту

Як видно на рис. 2.33 браузер має мінімальний захист від відбитка браузера, якого недостатньо, щоб уникнути відслідковування. Крім того:

- відсутнє блокування відстежувальної реклами;
- відсутнє блокування невидимих трекерів;
- відбиток браузера видається унікальним серед 236 890 перевірених за останні 45 днів;
- відбиток браузера передає щонайменше 17,85 біт ідентифікаційної інформації, тобто пристрій можна легко визначити серед інших.

Наступне тестування проводиться з налаштуваннями ПарашутOff за замовчуванням, тобто з середнім захистом, максимально комфортним для перегляду сторінок (рис.2.34).

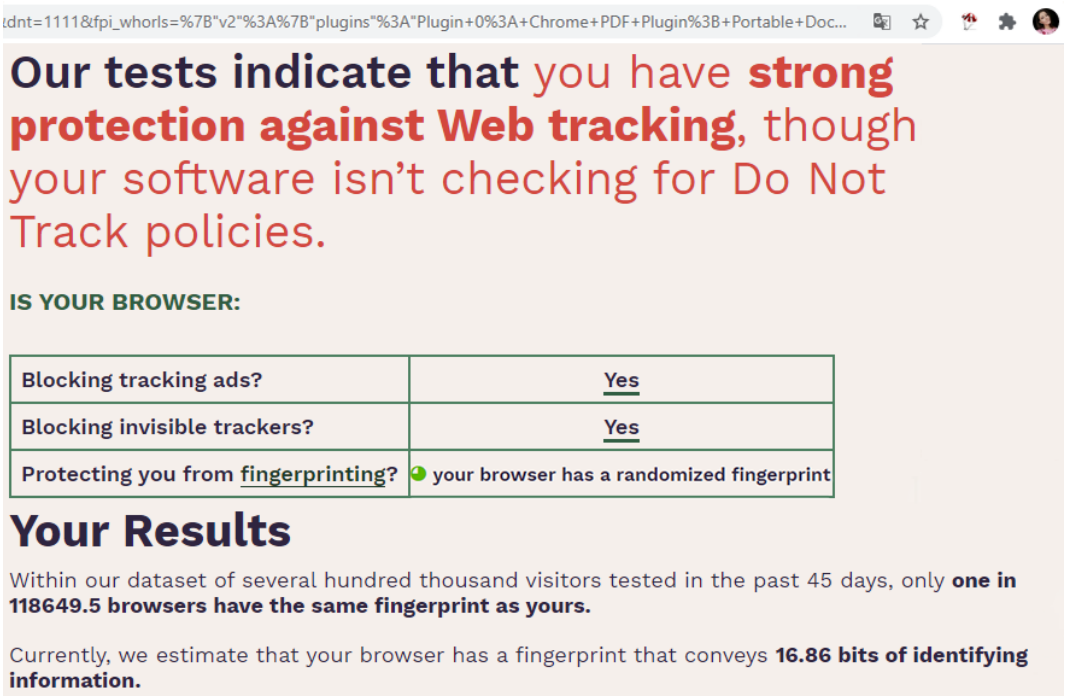


Рис. 2.34. Тестування після ввімкнення захисту

Як видно на рис. 2.34 тепер браузер має сильний захист від відбитка браузера, хоча й досі присутня проблема ігнорування DNTHeader. Крім того:

- з'явилося блокування відстежувальної реклами;
- з'явилося блокування невидимих трекерів;

- в межах набору даних декількох сотень тисяч відвідувачів сервісу Panopticlick, протестованих за останні 45 днів, один з 118 649,5 браузерів має той самий відбиток, що і відбиток протестованого, це набагато краще в порівнянні з попереднім тестом, де співпадінь взагалі не було знайдено;
- відбиток браузера передає лише 16,86 біт ідентифікаційної інформації, тобто пристрій уже не можна визначити стовідсотково.

Для кращого розуміння результатів слід порівняти, що саме змінилося.

Відмінність в значеннях рядка-ідентифікатора User-Agent показана на рис. 2.35. Розроблене розширення підмінило версію браузера Chrome з 86.0.4240.111 на більш часто використовувану користувачами 90.0.4430.212, чим і зменшило ентропію.

Рис. 2.35. Значення User-Agent: *а)* до ввімкнення захисту; *б)* після.

Відмінність в значеннях параметра Plugins показана на рис. 2.36. Розроблене розширення приховало інформацію про встановлені розширення, чим і зменшило ентропію.

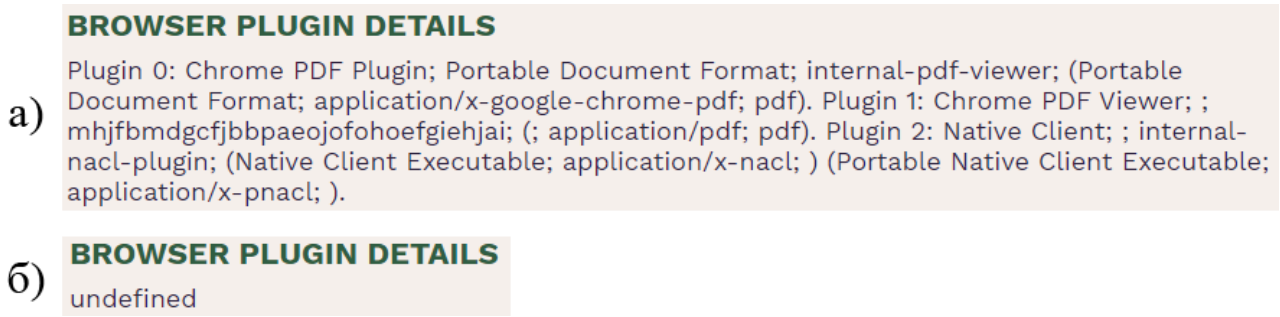
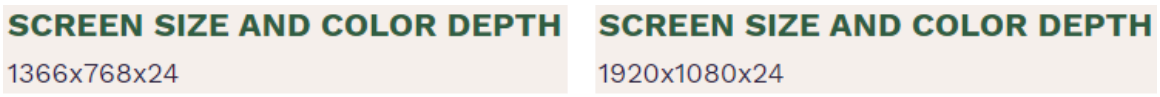


Рис. 2.36. Значення Plugins: *а)* до ввімкнення захисту; *б)* після.

Відмінність в значеннях розміру екрану та глибини кольору показана на рис. 2.37. Розроблене розширення підмінило 1366x768x24 на більш поширені 1920x1080x24, чим і зменшило ентропію.



а) б)

Рис. 2.37. Значення розміру екрану та глибини кольору: *а)* до ввімкнення захисту; *б)* після.

Відмінність в значеннях хешу відбитка Canvas показана на рис. 2.38. Розроблене розширення рандомізувало його, чим і зменшило ентропію.

а) б)

Рис. 2.38. Значення Canvas: *а)* до ввімкнення захисту; *б)* після.

Відмінність в значеннях хешу відбитка WebGL показана на рис. 2.39. Розроблене розширення рандомізувало його, чим і зменшило ентропію.

а) б)

Рис. 2.39. Значення хешу WebGL: *а)* до ввімкнення захисту; *б)* після.

Відмінність в значеннях Vendor і Renderer відбитка WebGL показана на рис. 2.40. Розроблене розширення приховало їх, чим і зменшило ентропію.

а)

б)

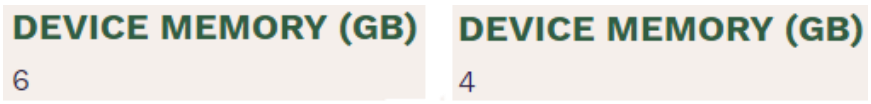
Рис. 2.40. Значення WebGL: *а)* до ввімкнення захисту; *б)* після.

Відмінність в значеннях кількості ядер ЦП показана на рис. 2.41. Розроблене розширення підмінило 4 ядра на 2, що більш часто зустрічається, чим і зменшило ентропію.

а) б)

Рис. 2.41. Значення кількості ядер ЦП: *а)* до ввімкнення захисту; *б)* після.

Відмінність в значеннях кількості гігабайт ОЗП показана на рис. 2.42. Розроблене розширення підмінило 6 Гб на 4 Гб, що більш часто зустрічається, чим і зменшило ентропію.



а) б)

Рис. 2.42. Значення кількості гігабайт ОЗП: *а)* до ввімкнення захисту; *б)* після.

Надані сервісом Panopticlick кількість бітів ідентифікаційної інформації і інформації, один із скількох браузерів має таке ж значення, що і протестований, для наочності зведено до табл. 2.1. Параметри, що не змінювались (часовий пояс, мова, клас ЦП), не включено до таблиці.

Таблиця 2.1

Порівняння значень отриманої від браузера інформації до і після ввімкнення захисту

Параметр	Значення до		Значення після	
	Біти ідентифікаційної інформації	Один із х браузерів має таке ж значення	Біти ідентифікаційної інформації	Один із х браузерів має таке ж значення
1	2	3	4	5
User-Agent	13,27	9892,96	2,75	26,9
Browser Plugin Details	3,14	8,83	0,82	1,76
Screen Size and Color Depth	3,97	15,67	1,46	5,52
Hash of Canvas	6,45	87,48	1,28	4,87

Продовження табл. 2.1

1	2	3	4	5
Hash of WebGL	10,97	2007,54	1,68	6,4
Webgl Vendor & Renderer	10,72	1692,07	1,37	5,16
Hardware Concurrency	2,59	6,0	2,46	5,51
Device Memory	11,16	2285,24	2,16	8,94
Разом:	62,27		13,98	

Отже, розроблене розширення ПарашутOff зменшило кількість ідентифікаційної інформації на 48,29 біт, тим самим зробивши ідентифікацію користувача майже неможливою.

2.4. Висновки до розділу 2

Враховуючи дослідження [33, 56] та результати аналізу методів захисту від відбитка браузера в розділі 1, як власне рішення вирішено запропонувати розширення для браузера Google Chrome, яке буде комплексним захистом від відбитка браузера.

Вивчення політики конфіденційності Google [57] показало, що браузер Google Chrome отримує величезну кількість інформації від користувача, але при цьому пропонує недосконалий захист, отже при користуванні ним необхідний додатковий захист від відбитка браузера.

Для реалізації власного рішення обрані такі засоби:

- середовище розробки програмного додатку WebStorm;
- фреймворк (програмний каркас) AngularJS;
- мови програмування:
  - CSS;
  - HTML;
  - JavaScript.

Для тестування обрано сервіс Panopticlick.

При розробці розширення було:

+ визначено місце розширення в схемі збору відбитків браузера;
+ написано маніфест – головний файл в Chrome-розширенні;
+ обрано іконки для розширення;
+ написано розмітку мовою HTML;
+ додано стилі, написані мовою CSS;
+ забезпечено функціонал сторінок за допомогою JavaScript;
+ щоб в майбутньому можна було перемикати мову розширення, створено окремий файл для повідомлень українською;
+ сформовано список шкідливих сайтів, які будуть блокуватися.

Для роботи з програмним продуктом потрібно:

- розпакувати розширення;
- зайти на панель «Розширення» Google Chrome;
- увімкнути режим розробника;
- завантажити розширення;
- ознайомитися зі встановленим розширенням.

Тестування показало, що розроблене розширення ПарашутOff зменшило кількість ідентифікаційної інформації на 48,29 біт, тим самим зменшивши ймовірність ідентифікації користувача.

ВИСНОВКИ

Інтернетом користуються мільярди людей, і їх безпека напряму залежить від ПЗ, що використовується для доступу до веб-сайтів та їх перегляду – веб-браузера. Але користувачі часто нехтують налаштуваннями безпеки, в зв’язку з чим зростає загроза атак, які використовують вразливості веб-браузерів, такі атаки останнім часом стали популярним способом зловмисників компрометувати комп’ютерні системи.

Серед розглянутих вразливостей браузера, одним з найголовніших виявилось зчитування відбитків браузера, оскільки загроза порушення конфіденційності – основна причина, через яку користувач повинен бути уважний . Відбиток браузера майже не змінюється з часом, що робить ідентифікацію практично стовідсотковою, він збирає величезний обсяг інформації. Крім того налаштування захисту від інших вразливостей теж входить до відбитка браузера. Зловмисник може використовувати дані про пристрій для застосування експлойтів чи «цифрового двійника», що несе значний збиток користувачу.

На сьогоднішній день гарантовано дієвих методів і засобів захисту від технології зчитування відбитка браузера не розроблено, однак після аналізу заходів, застосування яких дозволяє радикально знизити унікальність веб-браузера, одним з кращих виявилося застосування спеціалізованих розширень.

Як власне рішення запропоновано розширення для браузера Google Chrome, яке буде комплексним захистом від відбитка браузера.

Вивчення політики конфіденційності Google показало, що браузер Google Chrome отримує величезну кількість інформації від користувача, але при цьому пропонує недосконалий захист, отже при користуванні ним необхідний додатковий захист від відбитка браузера.

Для реалізації власного рішення обрані такі засоби:

- середовище розробки програмного додатку WebStorm;
- фреймворк (програмний каркас) AngularJS;

- мови програмування:
  - CSS;
  - HTML;
  - JavaScript.

Для тестування обрано сервіс Panopticlick.

При розробці розширення було:

+ визначено місце розширення в схемі збору відбитків браузера;
+ написано маніфест – головний файл в Chrome-розширенні;
+ обрано іконки для розширення;
+ написано розмітку мовою HTML;
+ додано стилі, написані мовою CSS;
+ забезпечено функціонал сторінок за допомогою JavaScript;
+ щоб в майбутньому можна було перемикати мову розширення, створено окремий файл для повідомлень українською;
+ сформовано список шкідливих сайтів, які будуть блокуватися.

Для роботи з програмним продуктом потрібно:

- розпакувати розширення;
- зайти на панель «Розширення» Google Chrome;
- увімкнути режим розробника;
- завантажити розширення;
- ознайомитися зі встановленим розширенням.

Тестування показало, що розроблене розширення ПарашутOff зменшило кількість ідентифікаційної інформації на 48,29 біт, тим самим зменшивши ймовірність ідентифікації користувача.

Отже, результатом виконаної роботи став модуль захисту конфіденційності користувача за допомогою налаштувань відбитка браузера. Під час виконання роботи було:

- проведено аналіз стану захищеності веб-браузера, що дало змогу зробити висновок щодо наявних вразливостей та загроз;

- проаналізовано існуючі методи захисту від відбитка браузера, що дозволило зробити обґрунтований вибір оптимального методу;
- запропоновано власне рішення проблеми, яке полягає у створенні національного продукту, а саме розширення ПарашутOff для браузера Google Chrome, з використанням засобів мов програмування JavaScript, HTML, CSS, фреймворка AngularJS та середовища розробки WebStorm, шляхом додавання нових засобів захисту від відслідковування, що дало змогу зменшити кількість інформації, зчитуваної відбитком браузера;
- проведено тестування модуля, що дало змогу переконатися в дієвості запропонованого рішення;
- розроблено рекомендації щодо використання модуля захисту.

Розроблене розширення є готовим продуктом, який може використовуватися як рядовими користувачами, так і комерційними організаціями. Розширення дозволить убезпечити комп’ютерну систему від цільової реклами та попередити зловмисницькі дії щодо витоку конфіденційної інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Simon Kemp. Digital 2021: Global Overview Report [Електронний ресурс] / Simon Kemp // DataReportal. – 2021. – Режим доступу: World Wide Web. – URL: https://datareportal.com/reports/digital-2021-global-overview-report.

2. What is a web browser? [Електронний ресурс] // Mozilla – Режим доступу: World Wide Web. – URL: https://www.mozilla.org/en-US/firefox/browsers/what-is-a-browser/.

3. Актуальные киберугрозы: итоги 2020 года [Електронний ресурс] // Positive Technologies. – 2021. – Режим доступу: World Wide Web. – URL: https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2020/.

4. Internet browser vulnerability and user security [Електронний ресурс] // Augusta Free Press. – 2020. – Режим доступу: World Wide Web. – URL: https://augustafreepress.com/internet-browser-vulnerability-and-user-security/.

5. The Analysis of Technologies Protecting from Web Browsers Identification [Електронний ресурс] / V. Moskovchenko та ін. – 2018. – Режим доступу: World Wide Web. – URL: https://doi.org/10.15688/nbit.jvolsu.2018.1.6

6. Yinzhi Cao. (Cross-)Browser Fingerprinting via OS and Hardware Level Features [Електронний ресурс] / Yinzhi Cao, Song Li, Erik Wijmans – Режим доступу: World Wide Web. – URL: http://yinzhicao.org/TrackingFree/crossbrowsertracking_NDSS17.pdf.

7. Peter Eckersley. How Unique Is Your Web Browser? [Електронний ресурс] / Peter Eckersley // Electronic Frontier Foundation, – Режим доступу: World Wide Web. – URL: https://coveryourtracks.eff.org/static/browser-uniqueness.pdf.

8. Про основні засади забезпечення кібербезпеки України [Електронний ресурс] // Офіційний вебпортал парламенту України. – 2017. – Режим доступу: World Wide Web. – URL: https://zakon.rada.gov.ua/laws/show/2163-19#Text.

9. Оксентюк Р. Використання інтернет-браузерів як засіб комунікації в управлінні інформаційними зв’язками [Електронний ресурс] / Роман Оксентюк

// Тернопільський національний технічний університет імені Івана Пулюя, Україна – Режим доступу: World Wide Web. – URL: http://elartu.tntu.edu.ua/bitstream/lib/21170/2/SEIED_2017_Oksentyuk_R-Internet_browser_as_a_method_58-60.pdf.

10. Кибербезопасность 2019-2020. тренды и прогнозы [Електронний ресурс] // Positive Technologies. – 2019. – Режим доступу: World Wide Web. – URL: https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-2019-2020/.

11. Способы проникновения вредоносных программ в систему [Електронний ресурс] // Энциклопедия «Касперского». – Режим доступу: World Wide Web. – URL: https://encyclopedia.kaspersky.ru/knowledge/how-malware-penetrates-systems/.

12. HTML & CSS [Електронний ресурс] // World Wide Web Consortium (W3C). – Режим доступу: World Wide Web. – URL: https://www.w3.org/standards/webdesign/htmlcss.html.

13. What is a URL? [Електронний ресурс] // MDN Web Docs. – Режим доступу: World Wide Web. – URL: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_URL.

14. HTTP vs HTTPS: What is Difference Between HTTP and HTTPS? [Електронний ресурс] // Guru99 – Режим доступу: World Wide Web. – URL: https://www.guru99.com/difference-http-vs-https.html.

15. Understanding Your Computer: Web Browsers [Електронний ресурс] // CISA. – 2019. – Режим доступу: World Wide Web. – URL: https://us-cert.cisa.gov/ncas/tips/st04-022.

16. Web History [Електронний ресурс] // Internet Archive – Режим доступу: World Wide Web. – URL: https://web.archive.org/web/20100925204436/http://www.w3c.rl.ac.uk/primers/history/origins.html.

17. Desktop, Tablet & Console Browser Market Share Worldwide. [Електронний ресурс] // StatCounter – Режим доступу: World Wide Web. – URL:

https://gs.statcounter.com/browser-market-share/desktop-tablet-console/worldwide/#monthly-202101-202105-bar.

18. Dave Roos. How ActiveX for Animation Works [Електронний ресурс] / Dave Roos // HowStuffWorks – Режим доступу: World Wide Web. – URL: https://entertainment.howstuffworks.com/activex-for-animation1.htm.

19. CVE-2020-7850 Detail [Електронний ресурс] // National Vulnerability Database – Режим доступу: World Wide Web. – URL: https://nvd.nist.gov/vuln/detail/CVE-2020-7850.

20. Results [Електронний ресурс] // National Vulnerability Database – Режим доступу: World Wide Web. – URL: https://nvd.nist.gov/vuln/search/results?form_type=Basic&results_type=overview&query=ActiveX&search_type=all.

21. Java Tutorial. [Електронний ресурс] // Java point. – Режим доступу: World Wide Web. – URL: https://www.javatpoint.com/java-tutorial.

22. Definition of Java sandbox. [Електронний ресурс] // PCMAG. – Режим доступу: World Wide Web. – URL: https://www.pcmag.com/encyclopedia/term/java-sandbox.

23. Results [Електронний ресурс] // National Vulnerability Database – Режим доступу: World Wide Web. – URL: https://nvd.nist.gov/vuln/search/results?form_type=Basic&results_type=overview&query=java+applets&search_type=all.

24. CVE-2020-14803 Detail [Електронний ресурс] // National Vulnerability Database – Режим доступу: World Wide Web. – URL: https://nvd.nist.gov/vuln/detail/CVE-2020-14803.

25. An Introduction to JavaScript [Електронний ресурс] // JavaScript.Info. – Режим доступу: World Wide Web. – URL: https://javascript.info/intro.

26. VBScript Tutorial [Електронний ресурс] // Tutorialspoint – Режим доступу: World Wide Web. – URL: https://www.tutorialspoint.com/vbscript/index.htm.

27. Jason Rafail. Cross-Site Scripting Vulnerabilities [Електронний ресурс] / Jason Rafail // Carnegie Mellon University – Режим доступу: World Wide Web. – URL: https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=52451.

28. Securing Your Web Browser [Електронний ресурс] // CISA. – Режим доступу: World Wide Web. – URL: https://us-cert.cisa.gov/publications/securing-your-web-browser.

29. Same-origin policy [Електронний ресурс] // MDN Web Docs. – Режим доступу: World Wide Web. – URL: https://developer.mozilla.org/ru/docs/Web/Security/Same-origin_policy.

30. Search [Електронний ресурс] // Carnegie Mellon University. – Режим доступу: World Wide Web. – URL: https://www.kb.cert.org/vuls/search/?q=cross-domain.

31. Browsing Safely: Understanding Active Content and Cookies [Електронний ресурс] // CISA. – 2019. – Режим доступу: World Wide Web. – URL: https://us-cert.cisa.gov/ncas/tips/ST04-012.

32. Бунин О. Browser Fingerprint – анонимная идентификация браузеров [Електронний ресурс] / Бунин О. // Хабр. – Режим доступу: World Wide Web. – URL: https://habr.com/ru/company/oleg-bunin/blog/321294/.

33. Мы проанализировали 500 000 цифровых отпечатков браузера, и вот что удалось обнатужить [Електронний ресурс] // ClickFraud. – 2021. – Режим доступу: World Wide Web. – URL: https://clickfraud.ru/my-proanalizirovali-500-000-czifrovyh-otpechatkov-brauzera-i-vot-chto-udalos-obnaruzhit/.

34. Как жулики используют цифрового двойника, чтобы расплатиться вашей картой [Електронний ресурс] // KasperskyDaily. – Режим доступу: World Wide Web. – URL: https://www.kaspersky.ru/blog/digital-masks-card-fraud/22584/.

35. Семенов Ю. Обзор уязвимостей, некоторых видов атак и средств защиты [Електронний ресурс] / Семенов Ю. // BookItep – Режим доступу: World Wide Web. – URL: http://book.itep.ru/6/intrusion.htm.

36. Саломатин А. А., Исхаков А. Ю. Application of the integrated indicator of browser fingerprinting in the problem of adaptive authentication of access sub-

jects [Електронний ресурс] / Саломатин А. А., Исхаков А. Ю. // Информационные и математические технологии в науке и управлении. – 2020. – Режим доступу: World Wide Web. – URL: https://doi.org/10.38028/esi.2020.20.4.008.

37. Septimiu-Vlad Mocan. Browser Fingerprinting and You (What It Is, How It Works, How It Violates Your Privacy, and What You Can Do) [Електронний ресурс] / Septimiu-Vlad Mocan. // TechNadu. – 2020. – Режим доступу: World Wide Web. – URL: https://www.technadu.com/browser-fingerprinting/102454/.

38. Chameleon [Електронний ресурс] // GitHub. – Режим доступу: World Wide Web. – URL: https://github.com/sereneblue/chameleon.

39. CanvasBlocker [Електронний ресурс] // GitHub. – Режим доступу: World Wide Web. – URL: https://github.com/kkapsner/CanvasBlocker

40. Canvas Defender [Електронний ресурс] // Firefox Browser. – Режим доступу: World Wide Web. – URL: https://addons.mozilla.org/uk/firefox/addon/no-canvas-fingerprinting/.

41. User-Agent Switcher [Електронний ресурс] // Firefox Browser. – Режим доступу: World Wide Web. – URL: https://addons.mozilla.org/en-US/firefox/addon/uaswitcher/?src=gitlab.

42. Oracle VM VirtualBox [Електронний ресурс] // VirtualBox. – Режим доступу: World Wide Web. – URL: https://www.virtualbox.org/.

43. Download VMware Workstation Player [Електронний ресурс] // VMware. – Режим доступу: World Wide Web. – URL: https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html.

44. Download VMware Fusion Download VMware Fusion [Електронний ресурс] // VMware. – Режим доступу: World Wide Web. – URL: https://my.vmware.com/en/web/vmware/downloads/info/slug/desktop_end_user_computing/vmware_fusion/11_0.

45. Parallels Desktop 16. [Електронний ресурс] // Parallels. – Режим доступу: World Wide Web. – URL: https://www.parallels.com/products/desktop/.

46. Introduction to Hyper-V on Windows 10 [Електронний ресурс] // Microsoft. – 2018. – Режим доступу: World Wide Web. – URL: https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/.

47. Про інформацію [Електронний ресурс] // Офіційний вебпортал парламенту України. – 1992. – Режим доступу: World Wide Web. – URL: https://zakon.rada.gov.ua/laws/show/2657-12#Text.

48. Про захист інформації в інформаційно-телекомунікаційних системах [Електронний ресурс] // Офіційний вебпортал парламенту України. – 1994. – Режим доступу: World Wide Web. – URL: https://zakon.rada.gov.ua/laws/show/80/94-вр#Text.

49. Про захист персональних даних [Електронний ресурс] // Офіційний вебпортал парламенту України. – 2010. – Режим доступу: World Wide Web. – URL: https://zakon.rada.gov.ua/laws/show/2297-17#Text.

50. Про затвердження Правил забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах [Електронний ресурс] // Офіційний вебпортал парламенту України. – 2006. – Режим доступу: World Wide Web. – URL: https://zakon.rada.gov.ua/laws/show/373-2006-п#Text.

51. ДСТУ 3396.1-96. Захист інформації. Технічний захист інформації. Порядок проведення робіт [Електронний ресурс] – 1997. – Режим доступу: World Wide Web. – URL: https://tzi.com.ua/downloads/DSTU%203396.1-96.pdf.

52. НД ТЗІ 2.5-004-99. Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу [Електронний ресурс] – 1999. – Режим доступу: World Wide Web. – URL: https://tzi.ua/assets/files/%D0%9D%D0%94-%D0%A2%D0%97%D0%86-2.5-004-99.pdf.

53. НД ТЗІ 2.5-005-99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу [Електронний ресурс] – 1999. – Режим доступу: World Wide

Web. – URL: https://tzi.ua/assets/files/%D0%9D%D0%94-%D0%A2%D0%97%D0%86-2.5-005--99.pdf.

54. НД ТЗІ 2.5-010-03. Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу [Електронний ресурс] – 2003. – Режим доступу: World Wide Web. – URL: https://tzi.com.ua/downloads/2.5-010-03.pdf.

55. НД ТЗІ 2.5-010-03. НД ТЗІ 1.1-002-99. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу [Електронний ресурс] – 1999. – Режим доступу: World Wide Web. – URL: https://tzi.com.ua/downloads/1.1-002-99.pdf.

56. Desktop, Tablet & Console Browser Market Share Worldwide. [Електронний ресурс] // StatCounter – Режим доступу: World Wide Web. – URL: https://gs.statcounter.com/browser-market-share/desktop-tablet-console/ukraine/#monthly-202101-202104-bar.

57. Privacy Policy [Електронний ресурс] // Google. – Режим доступу: World Wide Web. – URL: https://policies.google.com/privacy.

58. Cover Your Tracks [Електронний ресурс] // Electronic Frontier Foundation. – Режим доступу: World Wide Web. – URL: https://coveryourtracks.eff.org/.

59. WebStormetBrains [Електронний ресурс] // JetBrains. – Режим доступу: World Wide Web. – URL: https://lp.jetbrains.com/webstorm-ide/?gclid=Cj0KCQjwnueFBhChARIsAPu3YkTDlDrj0xtBBLDL4ot4ag8xpNn6J4M1n-wvxf1OHo053pMaNlDnacIaAlPrEALw_wcB&gclsrc=aw.ds.

60. AngularJS [Електронний ресурс] // AngularJS – Режим доступу: World Wide Web. – URL: https://angularjs.org/.