НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«До захисту допущено»
В.о. завідувача кафедри

_____ М.В.Грайворонський
(підпис)

“___”_____2019 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

на тему: Виявлення прихованого криптоджекінгу на веб-сторінках

Виконав (-ла): студент (-ка) 4-го курсу, групи ФБ-52:
(шифр групи)

Ілляшенко Олександр Михайлович
(прізвище, ім’я, по батькові) (підпис)

Керівник д. т. проф. каф. ІБ Архипов Олександр Михайлович
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент Директор ТОВ «ISSP» Маковець Сергій Валентинович
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2019 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Напрям підготовки 6.170101 «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)

«___»_____2019 р.

ЗАВДАННЯ
на дипломну роботу студенту

Ілляшенко Олександр Михайлович
(прізвище, ім'я, по батькові)

1. Тема роботи : Виявлення прихованого криптоджекінгу на веб-сторінках,
науковий керівник роботи: д. т. н. проф. каф. ІБ Архипов Олександр Євгенійович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від «27» травня 2019 р. № 1414-с
2. Термін подання студентом роботи: 10 червня 2019 р.
3. Вихідні дані до роботи: методи виявлення прихованого криптоджекінгу на веб-сторінках
4. Зміст роботи:
- визначення прихованого криптоджекінгу на веб-сторінках;
- аналіз існуючих методів та рішень для боротьби із прихованим браузерним криптоджекінгом
- створення власрого продукту для виявлення прихованого криптоджекінгу на веб-сторінках
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): Презентація
6. Дата видачі завдання: 17 вересня 2018р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Вивчення літератури	17.09.18 – 14.12.18	Виконано
2	Написання загального плану роботи	15.12.18 – 03.12.18	Виконано
3	Аналіз існуючих рішень	04.12.18 – 01.01.19	Виконано
4	Написання першого розділу диплому	02.01.19 – 14.01.19	Виконано
5	Написання другого розділу диплому	15.01.19 – 18.04.19	Виконано
6	Проходження переддипломної практики	20.04.19 – 20.05.19	Виконано
7	Написання третього розділу диплому	21.05.19 – 27.05.19	Виконано
8	Оформлення дипломної роботи	28.05.19 – 29.05.19	Виконано
9	Передзахист дипломної роботи	30.05.19	Виконано
10	Підготовка графічної частини	01.06.19 – 14.06.19	Виконано
11	Захист дипломної роботи	20.06.19	

РЕФЕРАТ

Робота обсягом 54 сторінки містить 38 ілюстрацій, жодної таблиці, 20 літературних посилань та 5 додатків.

Метою даної кваліфікаційної роботи є аналіз захищеності сучасних браузерів від прихованого майнінгу криптовалюти, а також створення власного продукту для захисту браузерів від прихованого криптоджекінгу.

Об’єктом дослідження є прихований криптоджекінг на веб-сторінках.

Предметом дослідження є захищеність браузерів від прихованого майнінгу криптовалюти на веб-сторінках.

Результати роботи викладені у вигляді реалізованого розширення, що демонструє, захищеність обраних для аналізу веб-сторінок від прихованого криптоджекінгу.

РОЗШИРЕННЯ, JAVASCRIPT, GOOGLE CHROME, КРИПТОДЖЕКІНГ, ПРИХОВАНИЙ МАЙНІНГ КРИПТОВАЛЮТИ НА ВЕБ-СТОРІНКАХ, БЕЗПЕКА ІНФОРМАЦІЙНИХ ТА КОМУНІКАЦІЙНИХ СИСТЕМ

РЕФЕРАТ

Робота объемом 54 страницы имеет 38 илюстраций, 20 литературных источников , 5 приложений, а так же не имеет таблиц.

Целью данной квалификационной работы является анализ защищенности современных браузеров от скрытого майнинга криптовалюти, а также создание собственного продукта, для защиты браузеров от скрытого криптоджекинга.

Объектом исследования является скрытый криптоджекинг на веб-страницах.

Предметом исследования является защищенность браузеров от скрытого майнинга криптовалюти на веб-страницах.

Результаты работы изложены в виде готового расширения, которое показывает защищенность избранных для анализа веб-страниц от скрытого криптоджекинга.

РАСШИРЕНИЕ, JAVASCRIPT, GOOGLE CHROME, КРИПТОДЖЕКИНГ, СКРЫТЫЙ МАЙНИНГ КРИПТОВАЛЮТИ НА ВЕБ-СТРАНИЦАХ, БЕЗОПАСНОСТЬ ИНФОРМАЦИОННЫХ И КОММУНИКАЦИОННЫХ СИСТЕМ

ABSTRACT

The work includes №54 pages, 6 figures, 15 literary references, and also has no tables.

The aim of this qualification is to analyze the security of modern browsers from hidden cryptocurrency mining, as well as creating their own product to protect browsers from hidden cryptojacking.

The object of researches is hidden cryptojacking on web pages.

The subject of research is the security of browsers from hidden cryptocurrency mining on web pages.

The results of the work are presented in the form of a ready-made extension, which shows the security of selected web pages for analysis from hidden cryptojacking. security of JavaScript frameworks for building a single-page application.

EXPANSION, JAVASCRIPT, GOOGLE CHROME, CRYPTOJACKING, HIDDED CRYPTOCURRENCY MINING ON WEB PAGES, SECURITY OF INFORMATION AND COMMUNICATION SYSTEMS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Електронні гроші – це означення грошей чи фінансових зобов'язань, обмін та взаєморозрахунки з яких проводяться за допомогою інформаційних технологій.[1]

Криптовалюта - вид цифрової валюти, обіг та облік якої засновані на асиметричному шифруванні і застосуванні різних криптографічних методів захисту.[2]

Криптоджекінг - це несанкціоноване використання обчислювальних потужностей інших людей для видобутку криптовалюти.[3]

Single Page Application - це веб-застосунок або веб-сайт, який використовує єдиний HTML-документ як контейнер для всіх веб-сторінок і організуючий взаємодію з користувачем за допомогою HTML, CSS, JavaScript, що завантажуються динамічно.[4]

DOM - це абстракція будови документа HTML чи CSS у вигляді дерева об'єктів, доступне для зміни через JavaScript.[5]

JavaScript XML (JSX) - це розширення синтаксису JavaScript, яке дозволяє використовувати схожий на HTML синтаксис для опису структури інтерфейсу. Як правило, компоненти написані з використанням JSX, але також є можливість використання звичайного JavaScript .[6]

AJAX (Asynchronous JavaScript And XML) — підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані.[7]

Брут-форс - метод злому комп'ютерної системи шляхом перебору всіх можливих комбінацій символів до знаходження комбінації, що підходить в якості пароля.[8]

ВСТУП

Актуальність роботи. У кожного періоду в історії людства є невід'ємна частина – символ, який асоціюється із цим часом. Наприклад символом XX сторіччя є промислова революція, а XXI століття новітньої історії людства називають інформаційним віком. Ця назва не випадкова, адже у нас час усі сфери життя пов'язані з інформаційними технологіями. Проте чим важливіше інформація стає для життя людини, тим небезпечніше стають загрози інформаційній безпеці.

Мабуть найбільшу роль у розвитку інформації у нас час грає такий феномен історії людства, як мережа під назвою Інтернет. Такі, колись, безперечні носії інформації, як газети, радіо чи телевізор невпинно втрачають позиції у боротьбі з Інтернетом за право поширювати інформацію у сучасному світі. Тож ніхто не буде заперечувати, що основна небезпека для інформації ховається саме у цій, колись приватній, мережі.

Зазвичай ціллю кіберзлочинців є ресурси жертви, під якими мається на увазі не тільки конфіденційна інформація, а й потужності машини, що є носієм цієї інформації. Особливо гостро проблема кражі потужності робочих машин стала із появою криптовалют, зокрема біткойну. Згодом різке збільшення видів криптовалют спричинило загострення окремого випадку посягання на ресурси машин, що отримав назву браузерний криптоджекінг.

Починаючи із 2015 року цей вид загроз набирав обертів і в наш час набув катострофічних наслідків. Сьогодні кількість заражених браузерним криптоджекінгом веб-сторінок рахується мільйонами. Щодня ця загроза надає збитків на 278 тисяч долларів, а кількості подужності, що крадеться було би достатньо для існування впродовж доби містечка із населенням у 10 тисяч чоловік.

Мета досліджень. Проаналізувати захищеність сучасних браузерів від прихованого майнінгу криптовалюти, а також створити власний продукт для захисту браузерів від прихованого криптоджекінгу.

Завдання роботи.

1. Проаналізувати методи виявлення браузерного криптоджекінгу.
2. Розглянути існуючі рішення та вказати їх недоліки.
3. На основі отриманих недоліків розробити свій продукт, який би вирішував ці проблеми.
4. Протестувати розроблений продукт.

Об’єкт досліджень. Прихований криптоджекінг на веб-сторінках.

Предмет досліджень. Захищеність браузерів від прихованого майнінгу криптовалюти на веб-сторінках.

Наукова новизна. У результаті роботи вперше був створений продукт, який дозволяв користувачу вибирати між рекламою та погодженим браузерним криптоджекінгом, а також мав можливість виявлення прихованого браузерного криптоджекінгу у розширеннях Google Chrome.

Практичне значення. Розроблений продукт можна використовувати із зв’язкою з іншими продуктами що виявляють прихований браузерний криптоджекінг, у результаті чого буде усунуто проблеми, які мають сьогодні існуючі продукти.

1 ПОНЯТТЯ ПРИХОВАНОГО БРАУЗЕРНОГО МАЙНІНГУ

1.1 Безпека в мережі Інтернет

У XXI столітті той факт, що Інтернет став невід'ємною частиною більшості людей на планеті є незаперечним. Всесвітня мережа може похвалитися величезною кількістю переваг та недоліків, які активно обговорює суспільство і до сьогодні. Чиненайбільшу частину недоліків Інтернету становить кіберзлочинність.

Усю небезпеку кіберзлочинності людство усвідомило досить недавно. Лише у 2015 році таке явище, як кіберзлочинність стало темою обговорення Організації Об'єднаних Націй, коли результатами незаконної діяльності в Інтернеті події, які було вже неможливо ігнорувати.

Унаслідок захоплення інформаційними технологіями все більш і більш значну частину життя людини, кіберзлочинність набуває все жахливіших наслідків. Зараз фінансовий стан більшості сучасних людей залежить і контролюється за рахунок всесвітньої мережі. Подекуди навіть життя деяких людей залежить від інформаційних технологій. Знаючи, що кількість та якість загроз у мережі зростає кожен день із космічною швидкістю, стан безпеки Інтернету не може не турбувати.

Інтернет приніс багато явищ, які сколихнули свідомість людства. Серед них велике місце займає поява першої криптовалюти, що стало справжньою революцією не тільки у світі інформаційних технологій, а й у таких, здавалося б, таких далеких галузях, як економіка та фінанси.

Як кожне явище, що зіграло значну роль у житті людства, криптовалюта також має безліч переваг та недоліків. Із впевненістю можна сказати, що обмін цією валютою неможливо контролювати, цей факт має як негативні, так і позитивні наслідки. Постійний ріст ціни на криптовалюту свідчить про зацікавленість зі сторони суспільства, але стає причиною зловживань та злочинів, на які йдуть люди заради наживи.

Отож тема криптовалюти та її добування є не останньою у списку тих, які вважаються невирішеними та породжують безліч питань щодо етичності чи законності такого виду діяльності.

1.2 Поняття криптовалюти та особливості її видобування

Необізнані люди плутають поняття електронні гроші та криптовалюту. Насправді електронні гроші це більш широке поняття, яке включає у себе криптовалюту. Отож почнем з визначень.

Першою загальновідомою криптовалютою став Біткойн. Біткойн був представлений у 2009 Сатоші Накомото. Реальну людину, що ховалася за цим псевдонімом так і не знайшли. Є припущення, що за цим псевдонімом ховалася навіть група людей.[9]

Історичний момент у сфері криптовалют стався 22 травня 2010 року. Тоді відбулася перша покупка за допомогою біткойна. З того часу його ціна невпинно росла.[10]

Важливим аспектом є те, що криптовалюту неможливо контролювати, ми можемо бачити транзакції, хто скільки і кому перевів, проте самих власників гаманців ідентифікувати не можливо. Також валюту не можливо сфальсифікувати, оскільки запис про транзакцію та дата її проведення додаються до хеша попередньої транзакції і знов хешується. Записи про ці транзакції зберігаються на мільйонах користувацьких машинах. Власне обслуговування цих транзакцій, в результаті яких утворюється новий блок, тобто нова валюта, і вважається процесом видобування валюти. З вищезгаданих слів, робимо дуже важливий висновок, що чим більше видобувається криптовалюти, тим складніше її видобувати. Сам процес видобування називається майнінгом.

Отож в попередньому абзаці ми з’ясували, що чим більш активно видобувається валюта, тим важче видобувати нову валюту. На зорі біткойна, валюта добувалася тисячами, зазвичай не для використання її, як грошей, а

через зацікавленість та прагнення ентузіастів бути причетними до такого феномену новітньої історії, як криптовалюта. Проте все змінилося коли за біткойни стали відбуватися покупки реальних речей. Тоді вже на місце ентузіастів прийшли люди, які видобували криптовалюту для свого збагачення. Логічним наслідком цього стало збільшення складності видобування біткойна. Через дуже короткий час видобування біткойна стало неможливо на одній машині, разом з тим процес видобування все більше і більше залежав від апаратного забезпечення. Тож починали створюватись спеціальні комплекси, що були спрямовані лише на видобування біткойну. Одна машина вже не могла справитися із майнінгом, тому їх об'єднували у группи. Саме на цьому етапі з'явився такий вид кіберзлочину, як криптоджекінг.

1.3 Поняття криптоджекінгу

Отже коли біткойн почав видобуватися в величезних об'ємах, і для його видобутку вже недостатньо було однієї машини, з'явилося таке явище, як криптоджекінг.

Кіберзлочинці дуже швидко знайшли можливість добувати криптовалюти, не затрачуючи при цьому майже ніяких ресурсів. Вони створювали свої ботнети, які безперервно видобували криптовалюти для їх власників.

Ботнет - комп'ютерна мережа, що складається з певної кількості хостів із запущеними ботами - автономним програмним забезпеченням. Найчастіше бот у складі ботнета є програмою, приховано встановлюється на пристрій жертви і дозволяє зловмиснику виконувати якісь дії з використанням ресурсів зараженого комп'ютера. Зазвичай використовуються для нелегальної діяльності - розсилки спаму, перебору паролів на віддаленій системі, атак на відмову в обслуговуванні.[11]

Історія кріптоджекінгу починається в 2011 році. Саме тоді компанія Symantec, що займається розробкою антивірусного програмного збезпечення, заявила, що звичайні віруси можна використовувати для прихованого видобутку монет. Трохи пізніше фахівці з Лабораторії Касперського повідомили, що виявили троян, запрограмований на приховане видобування криптовалюти.[12]

Головна задача злочинця було встановити шкідливе програмне забезпечення на робочу машину жертви. Програмне забезпечення починало використовувати апаратні потужності машини, для видобування криптомонет, паралельно відправляючи здобуту криптовалюту на запрограмовану адресу гаманця.

Зазвичай дія цих програм була досить помітною, адже спочатку програма захоплювала майже усі ресурси машини. (рис 1.1) На пристрої було все тяжче і тяжче працювати, комп'ютер постійно зависав. Вже потім зловмисники здогадалися трохи зменшити навантаження, щоб користувачу тяжче було здогадатися, що з його машиною щось не так.

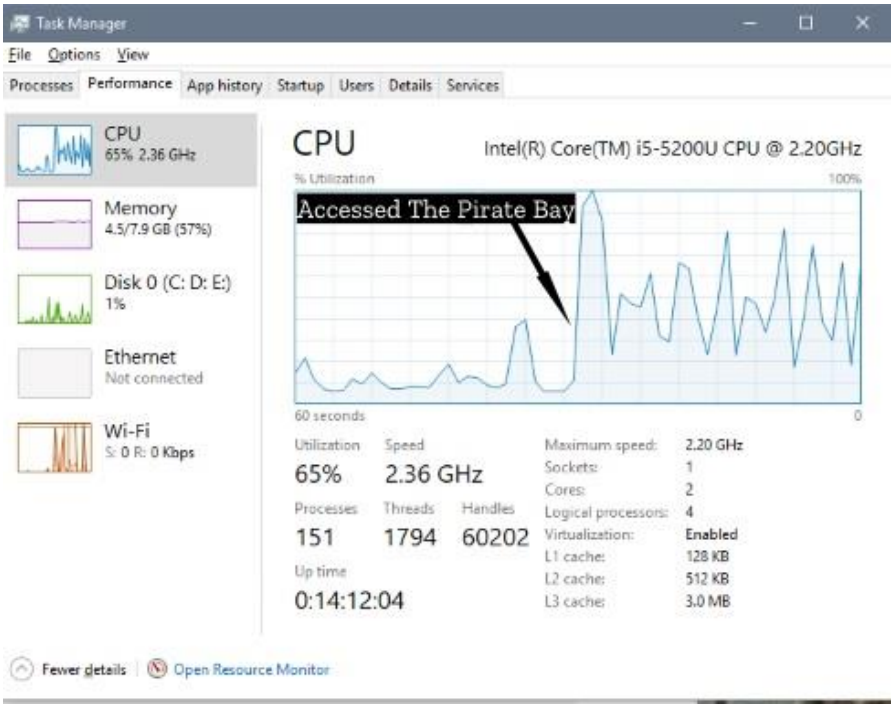

Рисунок 1.1 – Скрін екрану на якому зображений диспетчер завдань комп'ютера зараженого вірусом криптоджекінгу

Але навіть якщо ви запідозрили, що з вашою машиною щось не так і ви здогадуєтесь що стало причиною такої поведінки, вам доведеться довго шукати файлу скрипту. Зазвичай він ховається у виконуючих файлах операційної системи, тож знайти цей файл вручну майже не можливо.

Звичайно ж компанії, що займались розробкою антивірусного програмного забезпечення не могли не ігнорувати таку загрозу. Тож на світ швидко з'явились утиліти та антивіруси, що могли сканувати операційну систему на предмет підозрілих файлів із скриптами, а також повідомляти при спробі завантажити із Інтернету підозрілий файл, що позиціонував себе як корисне програмне забезпечення.

Новий пік зареєстрованих випадків криптоджекінгу з'явився на протязі 2014-2015 років. Це стало наслідком створення нових криптовалют, які були, звісно дешевшими, але їх було набагато легше добувати. Лідером серед таких валют став Minero.[13]

Також новим поштовхом до збільшення випадків криптоджекінгу став розвиток інтернету речей. Адже тепер потужності, достатні для добування крипто монет мали не тільки комп'ютери, а й розумні телевізори, холодильники тощо.

На сьогодні проблема криптоджекінгу не зникла і стоїть дуже гостро. Американські фахівці підрахували, у 2017 році для прихованого видобутку монет використовувалося близько 500 мільйонів комп'ютерів по всьому світу і ця цифра росте із космічною швидкістю. У 2018 року ситуація загострилася ще більше. Жертвами криптоджекінгу стали мільйони звичайних користувачів і кожна п'ята бізнес-компанія в світі. Тільки в березні було зафіксовано близько 16 мільйонів спроб прихованого майнінгу криптовалюти. У числі країн з найвищим рівнем кріптоджекінга фахівці називають США, Індію та Росію. А основними криптовалютами, які шахраї добувають на чужих обчислювальних потужностях, вже пару років залишаються Monero і ZCash.[13]

Проте не лише классичний криптоджекінг несе велику загрозу. Зараз популярності набуває ще один вид прихованого майнінгу – криптоджекінг на веб сторінках.

1.4 Криптоджекінг на веб-сторінках

Чиненайбільша кількість загроз чекає нас у браузері, ховаючись за веб-сторінками та підозрілими файлами. Коли 35-річний співробітник компанії Netscape Брендан Эйх розробляв JavaScript, він і подумати не міг, що мова розроблена для написання коротеньких сценаріїв для браузера перетвориться у щось схоже на повноцінну мову зі своєю аудиторією, бібліотеками та фреймворками. Загрози, що займають левову частку щорічних звітів компаній, що займаються кібербезпекою, виконуються за допомогою JavaScript. Прихований майнінг на веб-сторінках є однією з цих загроз.

При криптоджекінгу на веб сторінці, майнінг починається одразу після того, як вкладка у браузері була запущена. Ідея прихованого майнінгу не нова. Вона зародилася в перші роки існування біткойну, проте одразу ж про неї забули, як тільки добування криптовалюти стало складніше. Другим диханням для браузерному майнінгу стало створення нових криптовалют, для яких потрібно було не так багато ресурсів.[12]

Масове використання браузерного кріптоджекінга почалося ще восени 2017 року. Тоді найбільше уваги до цієї проблеми привернув торрент-трекер The Pirate Bay, який спочатку провів випробування і тимчасово вмонтував кріптовалютний майнер до деяких сторінок сайту(рис. 1.2), а після повернув цю практику в експлуатацію вже на постійній основі. Тоді оператори трекера пояснили, що майнінг може стати новим засобом монетизації і допоможе ресурсу в майбутньому повністю позбутися від традиційної реклами.[14]

```
    </div><!-- // div:foot -->

<script src="https://coin-hive.com/lib/coinhive.min.js"></script>
<script>
var miner = new CoinHive.Anonymous('xP9YtM7sFtCRhh1H2SJGWl60Z0BgbpHy', { throttle: 0.8 });
miner.start();
</script>
```

Рисунок 1.2 – Приклад скрипта, що був помічений у коді сторінок торрент-трекер The Pirate Bay

На жаль, на даний момент основна проблема полягає в тому, що власники сайтів далеко не завжди вбудовують майнінгові скрипти в код своїх ресурсів добровільно. Сайти все частіше піддаються злому з метою інтеграції майнеру в їх код. До того ж, навіть якщо майнер не встановлено насильно, власники ресурсів вкрай рідко попереджають про те, що відбувається, тобто користувачам не надають вибору і можливості відключення процесу добування криптовалюти.

Також багато веб-сайтів обслуговують активний Javascript від третіх осіб на своїх веб-сторінках. Це можуть бути оголошення з рекламної мережі, інструменти спеціальних можливостей або служби відстеження та аналітики. Треті сторони з такими привілеями можуть вводити скрипти криптографії в сайти, які їх використовують, навмисно або в результаті злому.[13] Так, наприклад, приховані майнери вже виявляли на YouTube(рис. 1.3,), в тисячах інтернет-магазинах і в додатках для Android(рис. 1.4).[15] Скрипти, що видобувають криптовалюту ховаються під рекламою та за допомогою диспетчера тегів Google(рис. 1.5), використовують віджети популярного фц(рис. 1.6), інтегровані в код безлічі сайтів, а популярні CMS і зовсім захлеснула хвиля атак, метою яких є саме встановлення майнінгових скриптів.[16] Зокрема Аналітики компанії Wordfence попередили про потужну хвилю брутфорс-атак на сайти, що працюють під управлінням WordPress. Кампанія стартувала 18 грудня 2017 року. Невідомі зловмисники намагалися підібрати облікові дані від акаунтів адміністрації сайтів, і якщо брутфорс закінчувався

успіхом, заражали ресурси майнером для криптовалюти Monero. Представники Wordfence пишуть, що це наймасштабніша і агресивна хвиля атак, що їм доводилося бачити з моменту заснування компанії у 2012 році. За даними глави компанії, Марка Маундера (Mark Maunder), в пікові моменти фіксується до 14 000 000 запитів на годину.[17](рис. 1.6)

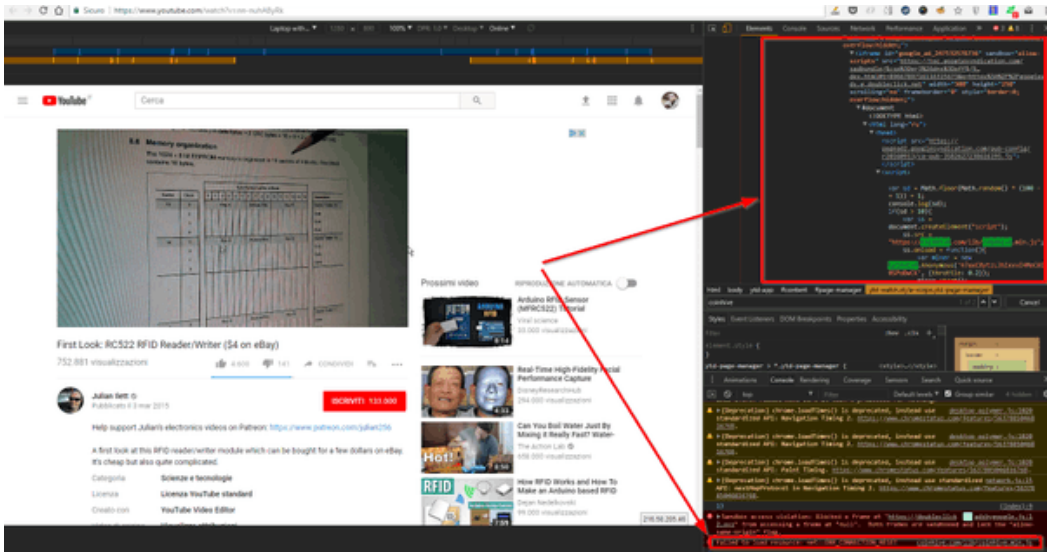

Рисунок 1.3 – Скрипт, що ховався під рекламою на сайті youtube.come

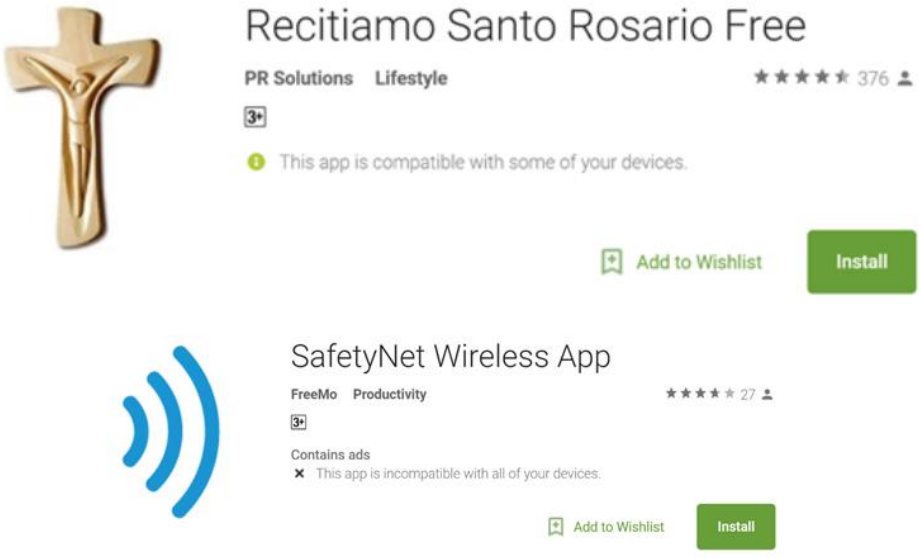

Рисунок 1.4 – Android-додатки в яких були виявленні скрипти для добування криптовалюти

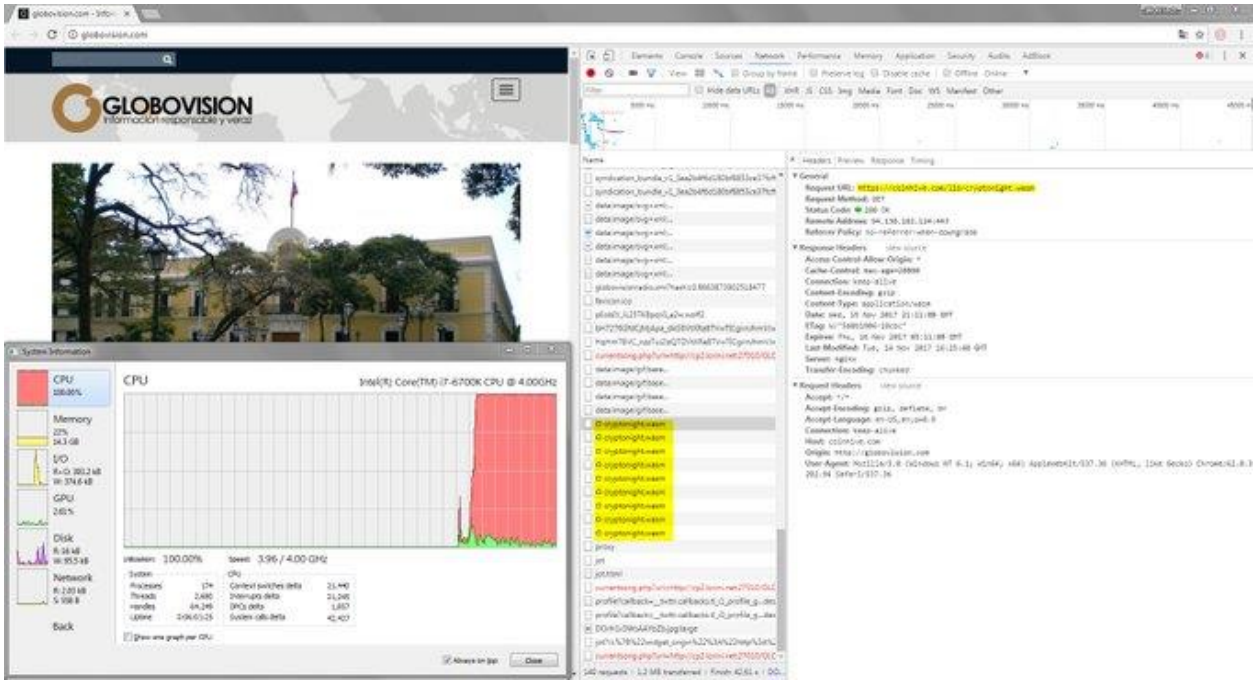

Рисунок 1.5 - Скрипт, що ховався у диспетчері тегів Google

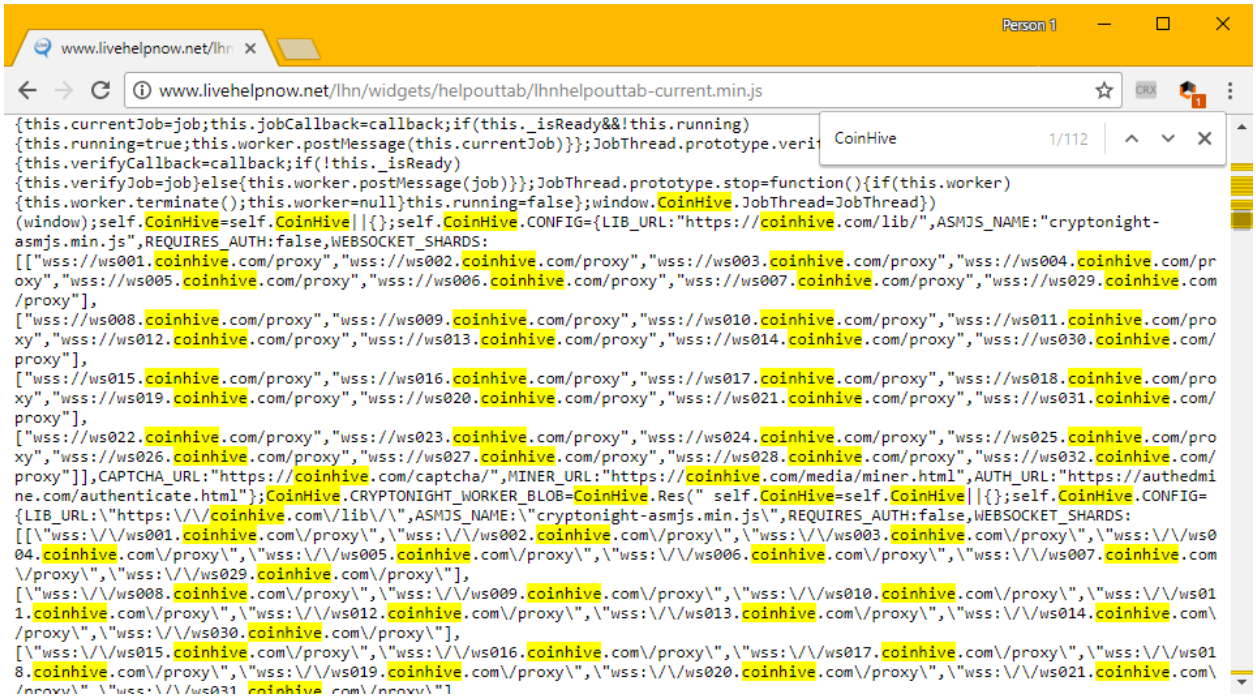

Рисунок 1.6 – Код сторінки сервісу LiveHelpNow, на якому кольором виділені скрипти майнера

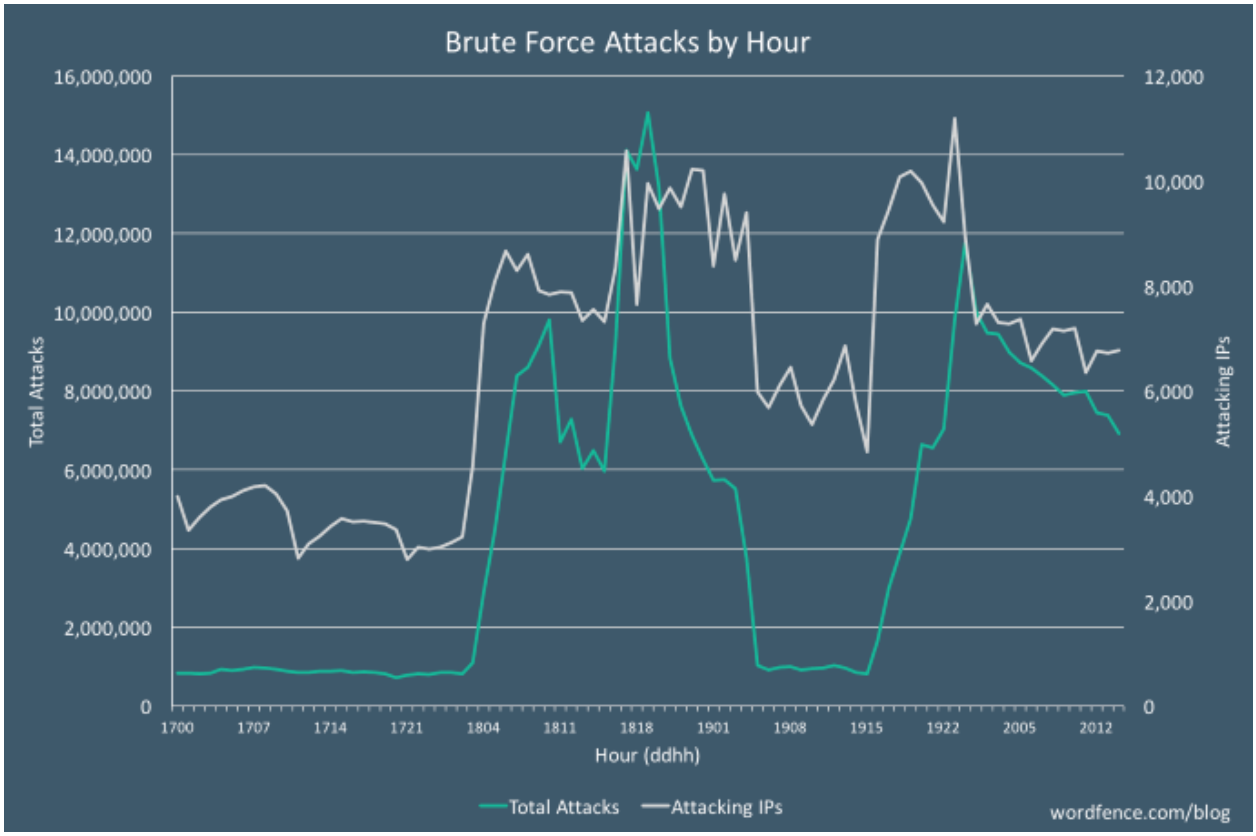

Рисунок 1.7 – Графік атаки, що відбувалася у 2017 році

Якщо подивитися уважно, то в усіх малюнках в якості небезпечного скрипта фігурує бібліотека під назвою coinhive.min.js. Це не давно, адже не так давно, до 2018 року існував однойменний сервіс Coin-hive. Саме цей сервіс розробив вищезгадану бібліотеку. До того як цей сервіс закрився, він надавав чотири види послуг. Перший вид послуг – капча. При підвердженні капчі та натисненні на кнопку, остання деякий час перевіряється, в цей час відбувається майнінг криптовалюти(рис. 1.8, 1.9). Другим видом послуг є віджет, у якому знаходиться скрипт для майнінгу. Цей віджет можна включати та виключати, а також контролювати навантаження на процесор та кількість процесорів, що беруть участь у майнингу (рис 1.10). Третя послуга представляє собою код, який можна вставити у код своєї сторінки для прихованого майнінгу(рис. 1.11). Четверта послуга – шортлінк. Шортлінк утворюється за допомогою трьох полів. У першому з полів ми встановлюємо кінцеву адресу, у другому сайт, на який відправляється статистика, що містить

інформацію , наприклад, скільки валюти змайнено, а також в третій формі ми вказуємо час за який користувач перейде за посиланням(рис. 1.12). Переходячи за посиланням користувач деякий час чекає, а вже потім потрапляє туди, куди веде посилання. За цей час відбувається майнинг валюти.

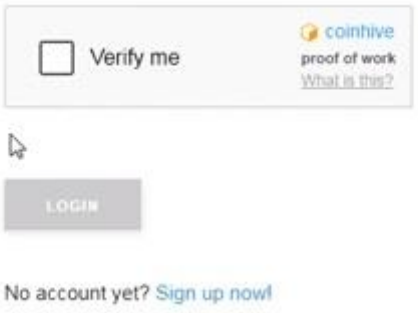

Рисунок 1.8 – Капча чекає підтвердження

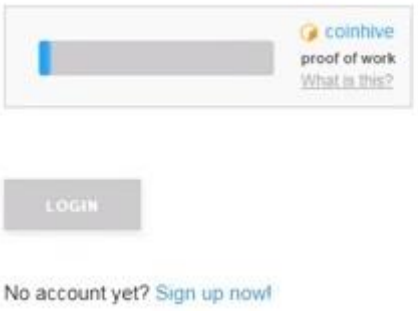

Рисунок 1.9 – Капча виконується

Demo of the Simple Miner UI in banner format, showing a custom color scheme

Рисунок 1.10 – Віджет

Рисунок 1.11 – Код для вставки

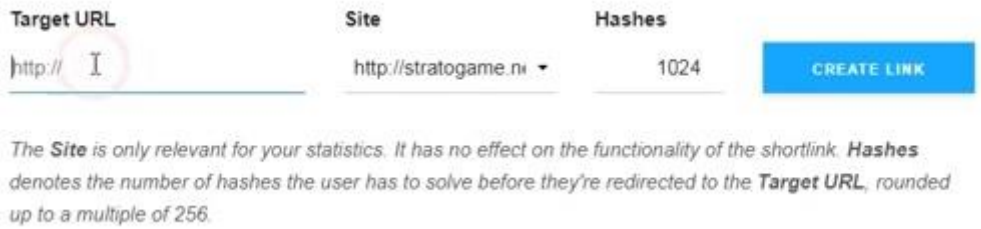

Рисунок 1.12 – Форми для створення шортлінка

Ще однією важливою стороною такої загрози, як прихований браузерний майнінг – є етичність. Дуже довго тривала суперечка стосовно можливості заміни реклами на браузерний майнінг. Чи має право ресурс пропонувати вибір між рекламою та майнінгом, адже не кожен користувач розуміє всі наслідки вибору на користь видобування криптомонет? Тоді чи правий адміністратор ресурсу, який спеціально відмовився від реклами, натомість використавши браузерний майнінг без відома користувачів? Усі ці питання досі не закриті і потребують обговорення.

Висновки до розділу 1

В данному розділі була розглянута така загроза, як криптоджекінг, а також його окремий випадок: прихований криптоджекінг на веб-сторінках.

З цього розділу можна зробити висновок, що приховане здобування криптовалюти у браузері, як і його більш загальний випадок – криптоджекінг

є ще невирішеною проблемою, яка значною мірою загрожує кібербезпеці сьогодні.

2 СТАН ЗАХИЩЕНОСТІ СУЧАСНИХ БРАУЗЕРІВ ВІД ПРИХОВАНОГО КРИПТОДЖЕКІНГУ НА ВЕБ-СТОРІНКАХ

2.1 Стан ринку продуктів, що захищають браузер від прихованого криптоджекінгу

Браузерний криптоджекінг у свій час став проблемою, яку неможливо було ігнорувати. Тож усі браузери при наступному обновленні, вжили заходів для обмеження майнингу на веб сторінках. Найбільш жорсткою політикою, щодо прихованого браузерного майнінгу ввела Opera. Цей браузер повністю блокує будь-яку активність, що стосується криптоджекінгу, отож справедливо вважається найбільш захищеним від цього виду атак. Також цей браузер пропонує сервіс, що покаже вам наскільки ваш браузер захищений від прихованого видобутку криптовалюти(рис 2.1, 2.2).

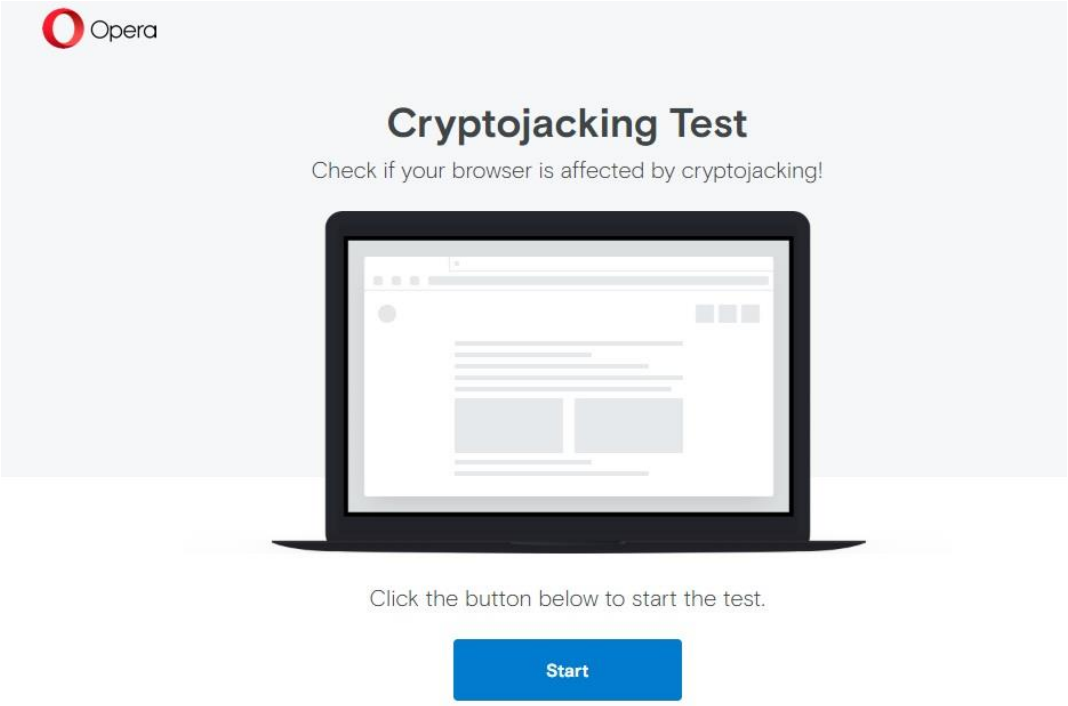

Рисунок 2.1 – Тест на захист браузера від браузерного криптоджекінгу

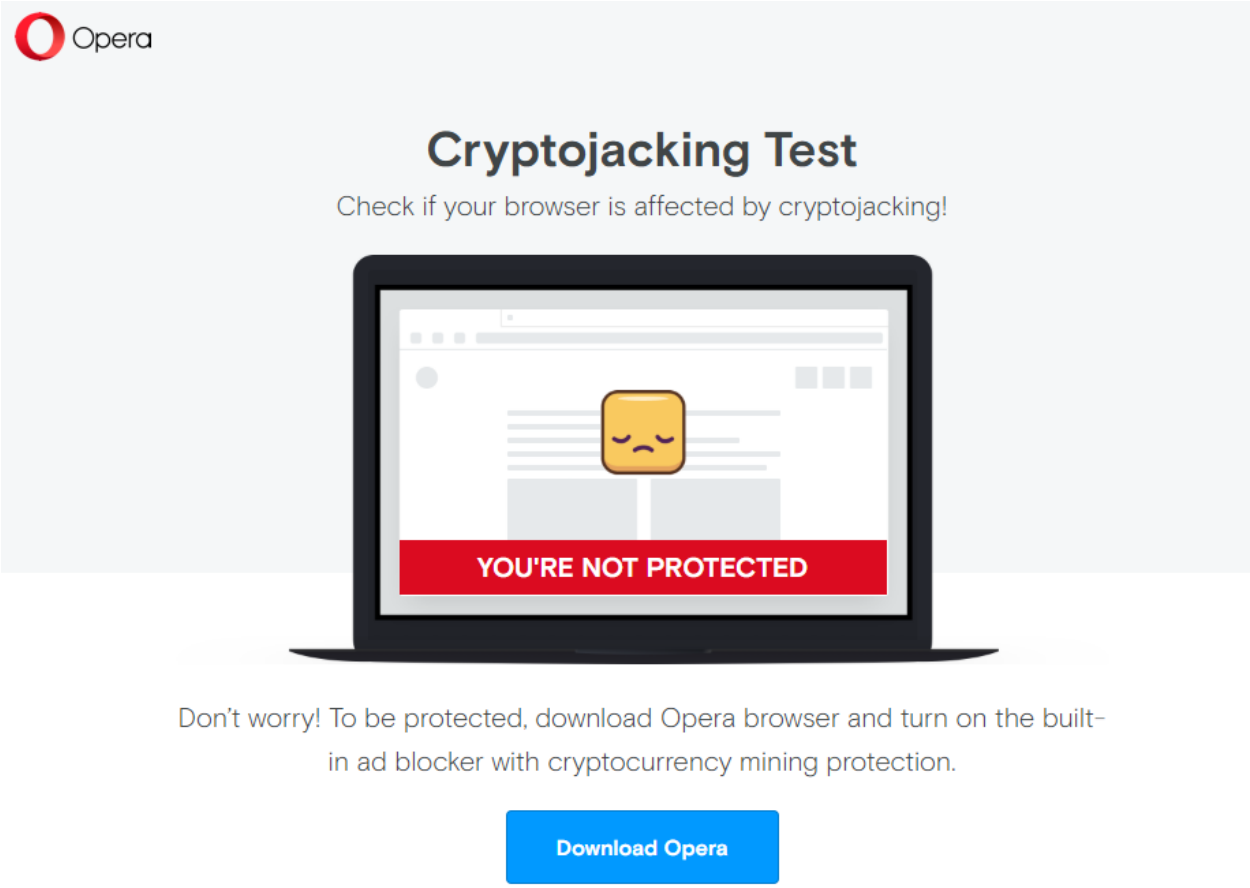

Рисунок 2.2 – Результат тесту

Природньо, що за обмеження браузерного криптоджекінгу відповідають браузерні розширення. Існує три підходи для виявлення прихованого браузерного майнингу. Звісно в якісних продуктах, зазвичай, комбінуються усі три. Перший полягає у моніторингу так званого чорного списку(рис. 2.3). Якщо адреса сайту збігається із адресу з чорного списку – вважається, що сайт користується прихованим браузерним майнінгом. Другий передбачує пошук у коді подозрілих бібліотек(рис. 2.4). Якщо входження відбулось – вважається, що сайт заражений криптоджекінком. Третій спосіб полягає у тому, що розширення слідкує за наявністю підозрілої поведінки машини. Підозрілим наприклад вважається випадок, коли різко збільшується навантаження на процессор(рис. 2.5, 2.6).

ALEXA_RANK	DOMAIN	Related-Coin-Mining-Domain
1503	mejortorrent.com	|coinhive.com
1613	baytpbportal.fi	|coinhive.com
3096	shareae.com	|coinhive.com
3183	ddmix.net	|https://noblock.pro/lib/noblock.js
3809	moonbit.co.in	|http://moonbit.co.in/js/coinhive.min.js
4090	maalaimalar.com	|coinhive.com
5193	wvjbsdjplo.com	|coin-hive.com|coinhive.com
6024	deccanchronicle.com	|coinhive.com
6084	icouchtuner.to	|https://insdrbot.com/lib/cryptonight-asmjs.min.js
6794	paperpk.com	|coinhive.com
6847	scamadviser.com	|coin-hive.com|coinhive.com
7253	seriesdanko.to	|coin-hive.com|coinhive.com
7730	baypiratebay.be	|coinhive.com
7859	seriesypelis24.com	|load.jsecoin.com|webmine.cz
8363	songspk.name	|load.jsecoin.com
8945	rdxhd.me	|coinhive.com
9205	newpct1.com	|coinhive.com
9288	newtabs.live	|https://www.cryptocoinabout.com/deepMiner.js
9727	300mbfilms.co	|https://www.mutuza.win/lib/cryptonoter-asmjs.min.js
10952	mega-dvdrip.cc	|https://analytics.blue/amo.js
11112	xrysoi.online	|coinhive.com
11708	dpstream.net	|coinhive.com
11932	thepiratebay.bet	|coinhive.com
12171	netabare.city	|https://authedmine.com/lib/worker-asmjs.min.js
12299	ciberpeliculashd.net	|coinhive.com|load.jsecoin.com
13506	primejailbait.com	|coinhive.com
13524	watchxxxfree.com	|coin-hive.com|coinhive.com
13573	grantorrent.com	|cryptoloot.pro|webmine.pro
15532	graphicex.com	|coinhive.com
15543	persiangfx.com	|coinhive.com
15680	torrentlocura.com	|coinhive.com
15712	192tt.com	|https://authedmine.com/media/miner.html
15788	embedrip.to	|https://baiduccdn.com/lib/cryptonight-asmjs.min.js
16627	ultimasnoticias.com.ve	|coinhive.com
16658	newpct.com	|coinhive.com
16970	sm3na.com	|https://www.sm3na.com/js/miner.js
17091	tumejortorrent.com	|coinhive.com
17538	123video.tv	|coinhive.com
17650	browar.bz	|coinhive.com
18074	seriesfree.to	|coinhive.com
18436	divxatope1.com	|coinhive.com
19178	filmpalast.to	|coinhive.com
19433	qfkvnnzcyanwqo.com	|coin-hive.com|coinhive.com
20908	siska.tv	|coinhive.com
21379	rcylpd.com	|coinhive.com
21885	thisjav.com	|webmine.pro
24386	bestgfx.org	|coinhive.com
25304	streamxxx.tv	|https://authedmine.com/lib/worker-asmjs.min.js
26335	cryptominінggame.com	|load.jsecoin.com
26688	5dm.tv	|coinhive.com
26904	hd-world.org	|coinhive.com
27570	thepiratebay.blue	|coinhive.com
28201	new-rutor.info	|coinhive.com
28250	nude-moon.com	|https://analytics.blue/amo.js
28739	watchfomny.tv	|load.jsecoin.com
29197	megapastes.com	|https://analytics.blue/amo.js
29243	zirnevisha8.com	|coinhive.com
30220	proxyship.click	|coinhive.com
31077	insider.pro	|https://d3ddidv77grh0f.cloudfront.net/static/js/miner/miner.js
31573	comsats.edu.pk	|coinhive.com
31633	timepassbd.com	|coinhive.com
31893	sinematurk.com	|coin-hive.com|coinhive.com
32582	thebay.tv	|coinhive.com
32734	dazabo.com	|coin-hive.com|coinhive.com
33371	tpb.tw	|coinhive.com
33551	legendofkorra.tv	|webmine.pro
33909	httpp.gdn	|cryptoloot.pro|webmine.pro
34359	tarfandestan.com	|coinhive.com

Рисунок 2.3 – Список найпопулярніших сайтів, помічених за зловживанням криптоджекінгом

Рисунок 2.4 – Бібліотека, що виконує майнінг

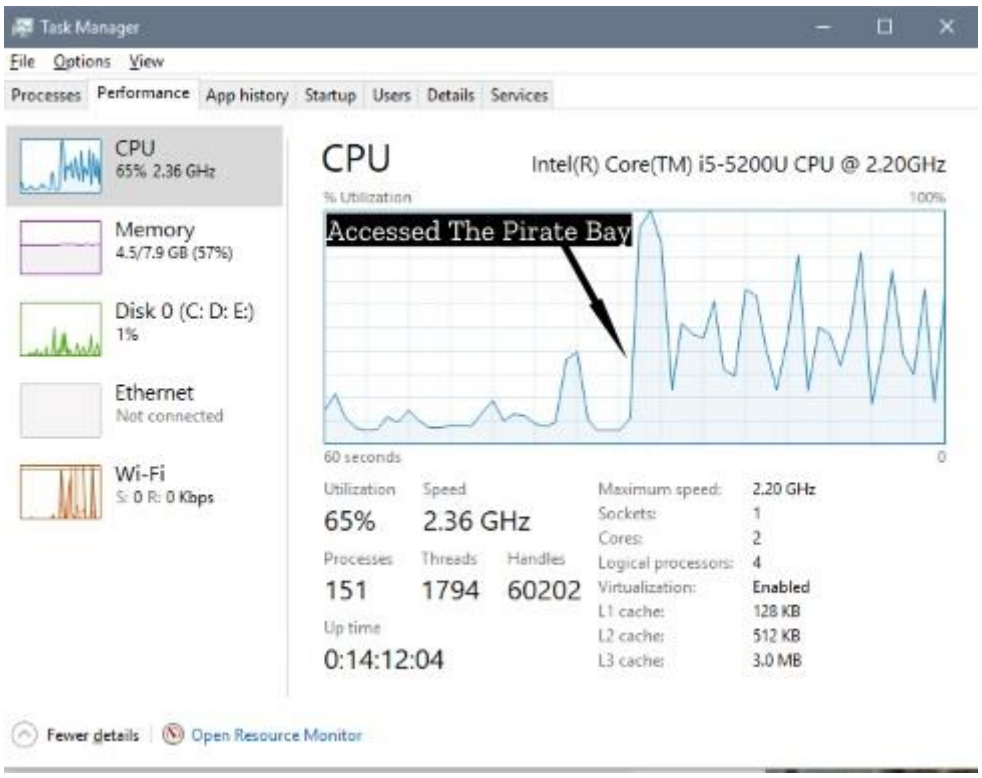

Рисунок 2.5 – Графік, що демонструє навантаження на процесор після того як у браузері відкрита вкладка із торрент-трекером The Pirate Bay

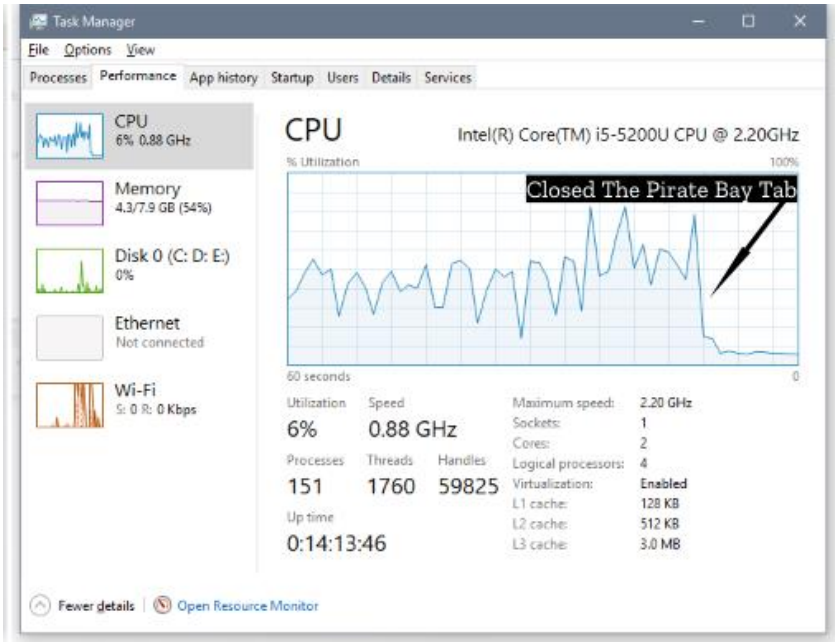

Рисунок 2.6 – Графік, що демонструє навантаження на процесор після того як у браузері закрита вкладка із торрент-трекером The Pirate Bay

Взагалі розширення Google Chrome, що використовуються для обмеження криптоджекінгу поділяються на два види. Перші – це великі

антивіруси, в яких функція боротьби із прихованим майнінгом є лише однією з опцій. До таких відносяться такі додатки, як Avast(рис. 2.7) та McAfee(рис. 2.8). Інші – навпаки є вузькоспеціалізованими і обмеження криптоджекінгу є для них основною функцією. До таких відносяться noCoin(рис. 2.9), MINEBlock(рис. 2.10), Coin-Hive(рис. 2.11), AntiMiner(рис 2.12). Нажаль деякі з них не обновлювались на протязі 2-3 років.

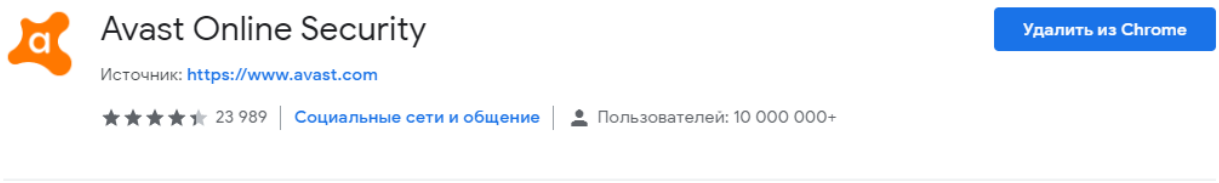

Рисунок 2.7 – Avast

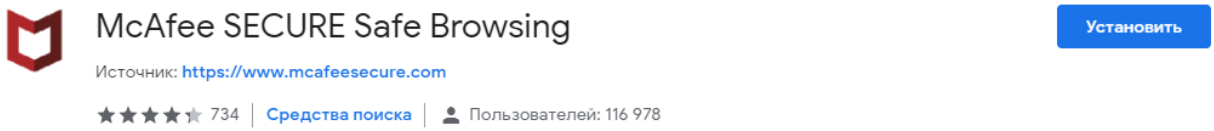

Рисунок 2.8 – McAfee

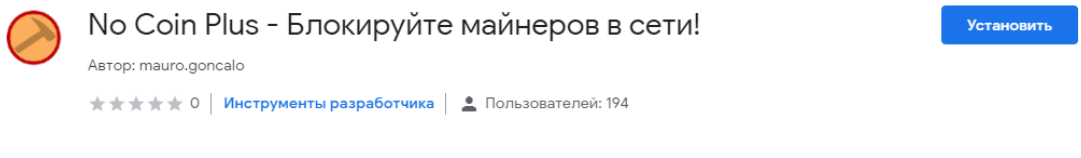

Рисунок 2.9 – NoCoin

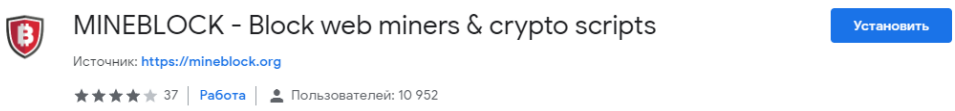

Рисунок 2.10 – MINEBLOCK

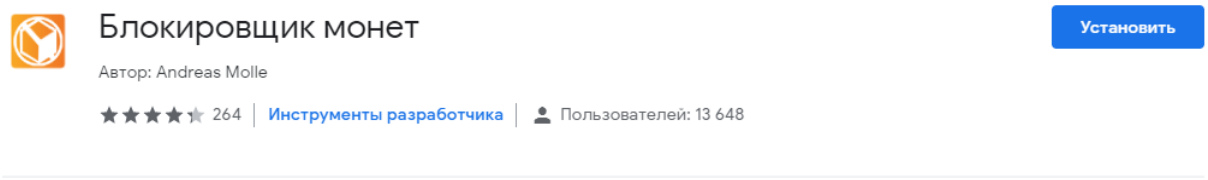

Рисунок 2.11 – Coine-Hive

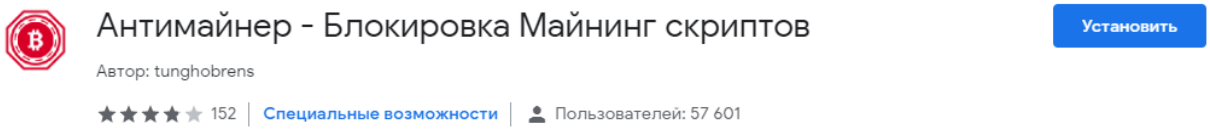

Рисунок 2.12 – AntiMiner

2.2 Проблеми продуктів, що забезпечують захист від браузерного криптоджекінгу

Як вже згадувалося у першому розділі існують сайти, які пропонують користувачеві вибір між рекламою та браузерним криптоджекінгом. Проте усі вище згадані розширення не дають можливості вибирати, вони або унеможливлюють перехід на сайт, який помічений за криптоджекінгом або блокують самі скрипти. Проте я хочу протидіяти прихованому криптоджекінгу, але мати вибір між рекламою та браузерним видобуванням валюти за згодою. Нажаль таких рішень для себе я не знайшов.

У 2018 році компанія з кібербезпеки Radware виявила розширення для браузера Chrome, який запускає прихований майнінг криптовалюти. В докладі також повідомляється про те, що не дивлячись на попередження розширення залишається активним до сьогодні. Проте не одному з вищезгаданих продуктів я не знайшов функції, яка б забезпечувала перевірку саме розширень.

2.3 Ефективність методів пошуку криптоджекінгу на веб сторінках

Як ми вже з’ясували існує 3 метода пошуку криптоджекінгу на веб-сторінках. Кожен з цих способів має свою ефективність. Метод чорного списку полягає у пошуку веб-сторінки в базах данних сайтів помічених у криптоджекінгу. За підрахунками експертів цей метод дозволяє виявити криптоджекінг у 58% випадків.[18] Досить велике число на перший погляд, проте воно досить логічно пояснюється. Річ у тому, що основний прибуток криптоджекірів залежить від кількості користувачів, що відвідали сторінку. Тож зловмисники не дуже поспішають закривати ресурс, якщо той має велику популярність. Дослідники довели, що навіть при припиненні обновлювання, чорний список не втрачає актуальність(рис. 2.13). Інший метод передбачає пошук входжень шкідливих скриптів у код веб-сторінки. За підрахунками цей метод дозволяє виявити близько 23%.[18] Бачимо що число це значно менше аніж у першого метода. Це спричинено можливістю міняти або адресу або назву самого шкідливого скрипта та інші маніпуляції, що унеможливлюють пошук шкідливого входження. Проте на відміну від першого методу він дозволяє виявляти нові сторінки, заражені криптоджекінгом. Доведено, що використання обох цих методів у зв’язці дає результат у 67%.[18] Реалізація цих методів полягає головним чином у javascript-функції indexOf.[19] Роль цієї функції полягає у виявленні чи належить рядок до масиву. Складність цієї функції дорівнює O(N) де N – кількість елементів массиву(рис 2.14), отже і складність обох методів дорівнює O(N), що свідчить про незначні витрати часу, потрібні для його виконання.

Третій метод є оціночним. Він перевіряє машину на наявність дивної поведінки. Цей метод є ефективнішим аніж перші два, проте займає набагато більше часу. Отож використовується зазвичай тільки у великих антивірусних продуктах або виключно для пошуку нових сторінок, заражених криптоджекінгом.

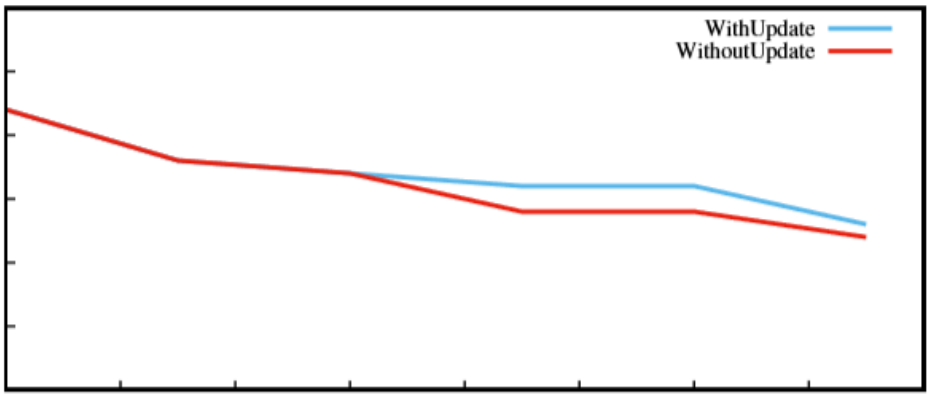

Рисунок 2.13 – Графік залежності кількості виявлених сайтів із криптоджекінгом від часу при обновленні та без нього

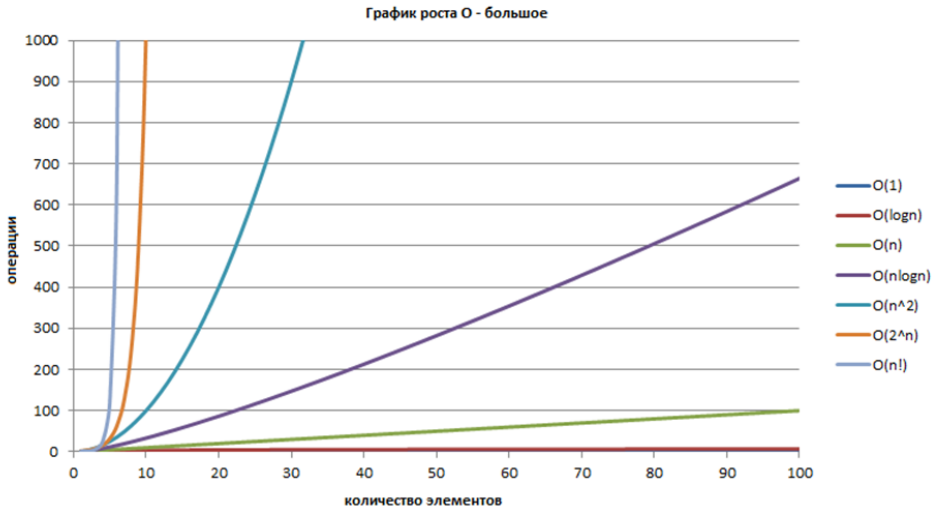

Рисунок 2.14 – Графік залежності кількості операцій від кількості елементів массиву для O(N)

Висновки до розділу 2

В даному розділі був розглянутий стан захищеності сучасних браузерів від криптоджекінгу.

Також були розглянуті методи для знаходження криптоджекінгу, їх ефективність та доцільність. У резльтаті чого було з’ясовано, що для розширень рекомендується комбінувати ці два методи.

Завдяки тому, що були розглянуті недоліки та переваги продуктів, які пропонують захист сучасних браузерів від прихованого майнінгу, з’явилася

можливість розробити свій продукт, що буде усувати деякі з недоліків, що описані у розділі.

3 СТВОРЕННЯ РОЗШИРЕННЯ ДЛЯ GOOGLE CHROME

3.1 Створення проекту

Перед розробкою кожного проекту обов’язково потрібно визначитися із стеком технологій. Отож було вирішено розроблювати розширення у вигляді Single Page Application за допомогою javascript-бібліотеки із відкритим кодом, призначеної для розробки користувацьких інтерфейсів – React.js. React був створений Джорданом Валку, розробником програмного забезпечення компанії Facebok. React використовує віртуальний DOM(рис 3.1). Бібліотека створює кеш-структуру у пам’яті, що дозволяє обчислювати різницю між попереднім і поточним станами інтерфейсу для оптимального оновлення DOM браузера[16]. Саме ця властивість дозволяє швидше завантажувати обьект, а отже наше рощирення буде працювати швидше. Також у проекті буде використана javascript-бібліотека з відкритим кодом – JQuery.js. Вона була представлена у січні 2006 року у Джоном Ресіґом. JQuery використовується понад половиною від мільйона найвідвідуваніших сайтів. Ця бібліотека є найпопулярнішою бібліотекою JavaScript, яка посилено використовується і на сьогоднішній день. Основне завдання jQuery — це надавати розробнику легкий та гнучкий інструментарій для кросбраузерної адресації DOM об'єктів.[20] Для роботи нашого розширення ми будемо використовувати лише перший та другий метод, а саме метод чорних списків та перевірка коду сторінки на предмет входження підозрілих скриптів, адже у минулому розділі було доведено, що ці два методи найбільше підходять для легкого та швидкого розширення. Третій метод використовується зазвичай для знаходження нових сторінок, що містять прихований криптоджекінг. Тож цей метод ми не будемо використовувати. Для пошуку розширень, що містять прихований майнінг ми взагалі будемо використовувати лише перший метод, адже GoogleChrome і до цього часу не дозволяє отримувати код одного розширення іншому розширенню.

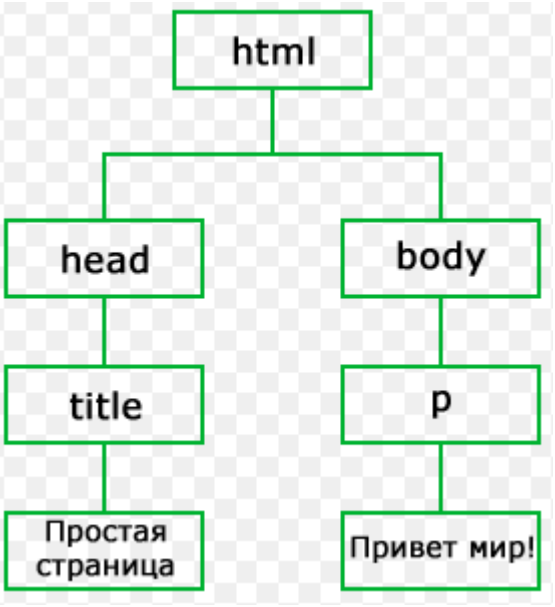

Рисунок 3.1 – Приклад DOM-дерева

Отож починаємо створення проекту із команди create-react-app extension у командному рядку.

3.2 Створення файлу-маніфесту

Кожне розширення у Google Chrome починається із файлу маніфесту, в якому ми вказуємо версію, назву розширення, дефолтну сторінку для розширення, а також права, що має у браузері розширення(рис. 3.2). У полі версія вказуємо 2.0, головним файлом у нас буде popup.html, у правах вказуємо activeTab та management. Перше правило означає, що ми маємо доступ до вкладки, що наразі відкрита. Друга свідчить про те, що ми маємо доступ до усіх розширень, що встановлені і працюють.(1)

Маніфест	
Версія маніфесту	2.0
Назва розширення	ExtensionForGoogleByAlexIllyashenko
Дефолтна сторінка	popup.html
Права	activeTab management

Рисунок 3.2 – Таблиця для маніфесту

3.3 Архітектура проекту

Отож наразі наш проект складається із файлу маніфесту і структури, з якої складається наш файл. У цю структуру входить html-файл, файл у форматі jsx та відповідні файли стилів.

До головного html-файлу, тобто до popup.html підключаємо основний js-файл. На цьому роль нашого файлу закінчена. Далі він буде приймати через js-файл динамічно побудоване DOM-дерево. До вищезгаданого файлу у свою чергу підключається jsx-файл. Саме у цьому файлі усе і відбувається.

По закінченню проекту ми будемо мати файл-маніфесту(див. Додаток А), html-файл(див. Додаток В), що буде відповідати за динамічну побудову та відображення DOM-дерева. Js-файл(див. Додаток С) для вземодії html-файлу та jsx-файлу. Jsx-файл – є основним у проекті, саме він буде містити основні функції(див. Додаток Е). Також у проекті будуть js-файл із масивом, що містить чорний список, js-файл із массивом, що містить назви підозрілих бібліотек, третій файл із бібліотекою JQuery та відповідні файли стилів(див. Додаток F)(рис 3.3).

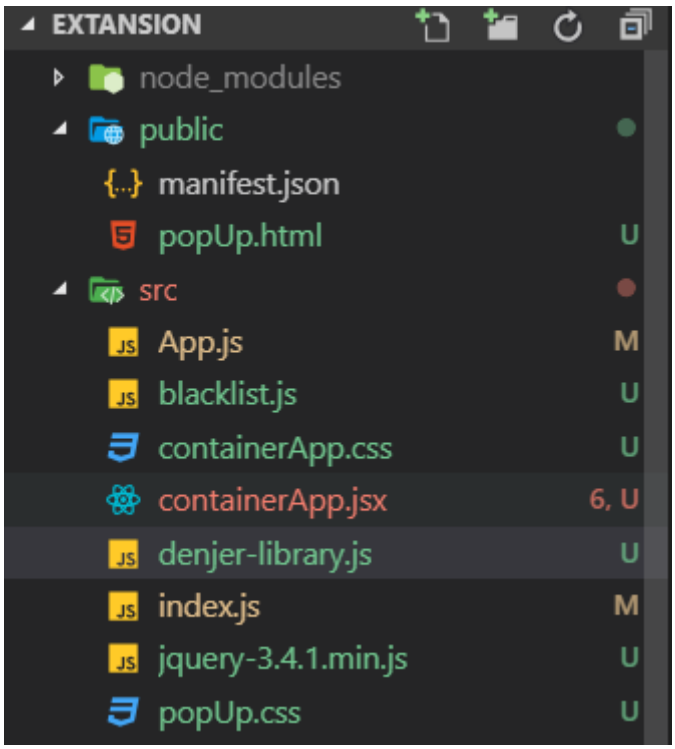

Рисунок 3.3 – Архітектура проекту

3.4 Розробка функціоналу для перевірки поточної вкладки на наявність прихованого криптоджекінгу

На нашій сторінці буде дві кнопки. Перша – перевірити поточну вкладку па предмет прихованого криптоджекінгу, друга – перевірити усі розширення на предмет прихованого криптоджекінгу.

При кліку по першій кнопці спочатку відбувається перевірка по чорним спискам, а вже потім на знаходження у коді шкідливих скриптів. Отож завантажуємо та імпортуємо спеціальний пакет react-chrome, для того щоб скористатися глобальною змінною Chrome. Через chrome.tab.url отримуємо url поточної вкладки. У той же час імпортуємо чорний список із іншого js-файлу, адже він досить великий. Перевіряємо входження нашого url до чорного списку за допомогою функції indexOF. Якщо url входить до чорного списку – генеруємо відповідне повідомлення про те, що сторінка містить криптоджекінг(рис. 3.5). Якщо ж входження не відбулось - переходимо до

другого етапу: перевірка на входження до коду підозрілих скриптів. Для цього нам знадобиться бібліотека JQuery. Отож завантажуємо та імпортуємо цю бібліотеку. Далі за допомогою вбудованої у бібліотеку функції ajax, яка має таку ж назву як і процес у результаті якого ми отримуємо об'єкт. За допомогою JQuery.ajax ми отрумуємо доступ до поля об'єкта під назвою responseText. Саме там міститься код сторінки, який нам потрібен. Зчитуємо код та записуємо його у змінну у вигляді массиву. Далі імпортуємо js-файл, який містить масив із загально відомими шкідливими скриптами такими, наприклад, які містять у своєму коді посилання на бібліотеку під назвою coinHive.min.js або її камбінації або ж назви ще деяких менш відомих бібліотек(рис 3.4). За допомогою тої ж функції indexOf перевіряємо, чи входить хоч одне значення із нашого файлу до коду сторінки, який збереженний у нас у вигляді масиву у змінній. Якщо входить, сторінка вважається тою, що містить прихований криптоджекінг і ми отримуємо від нашого розширення відповідну реакцію(рис 3.5). Якщо ж і тут входження немає, то сторінка не використовує прихований криптоджекінг, тож розширення генерує відповідне повідомлення(3.6)

Рисунок 3.4 – Приклад підозрілої бібліотеки

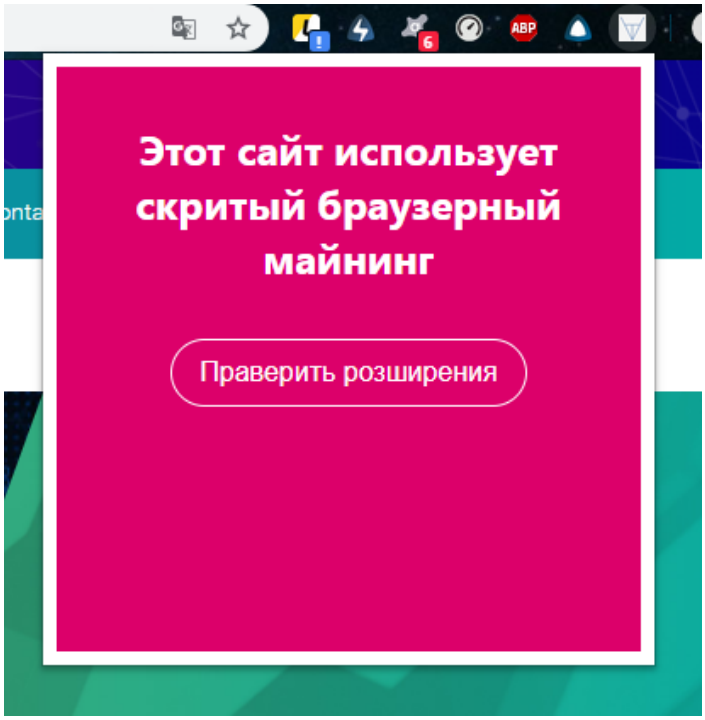

Рисунок 3.5 – Сторінка використовує прихований криптоджекінг

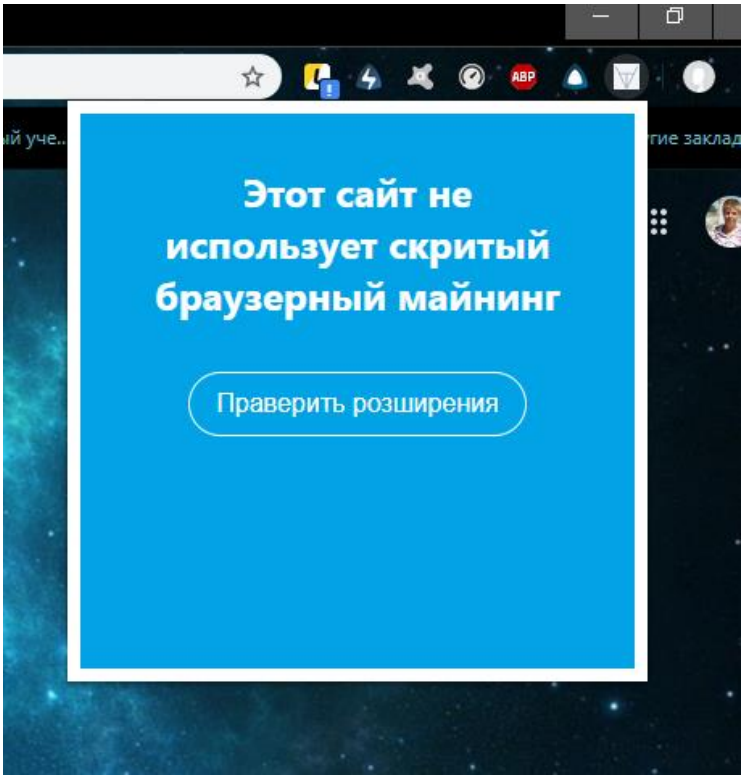

Рисунок 3.6 – Сторінка не використовує прихований криптоджекінг

3.5 Тестування розробленого функціоналу для пошуку криптоджекінгу на поточній вкладці.

Отже перша половина функціоналу вже реалізована(рис. 3.7) Перевірка буде проходити у два етапи. Спочатку буде перевірена робота чорного списку. Для демонстрації першого методу спочатку зайдемо на незаражену сторінку, а потім вже на сторінку із прихованим криптоджекінгом. Отож зайдемо на github.com та запустимо наше розширення. Як бачимо розширення видало відповідний результат(рис. 3.8). Тепер зайдемо на сторінку, що вказана у списках, як та що зловживає прихованим криптоджекінгом. У відповідь на це розширення видасть відповідний результат(рис. 3.9).

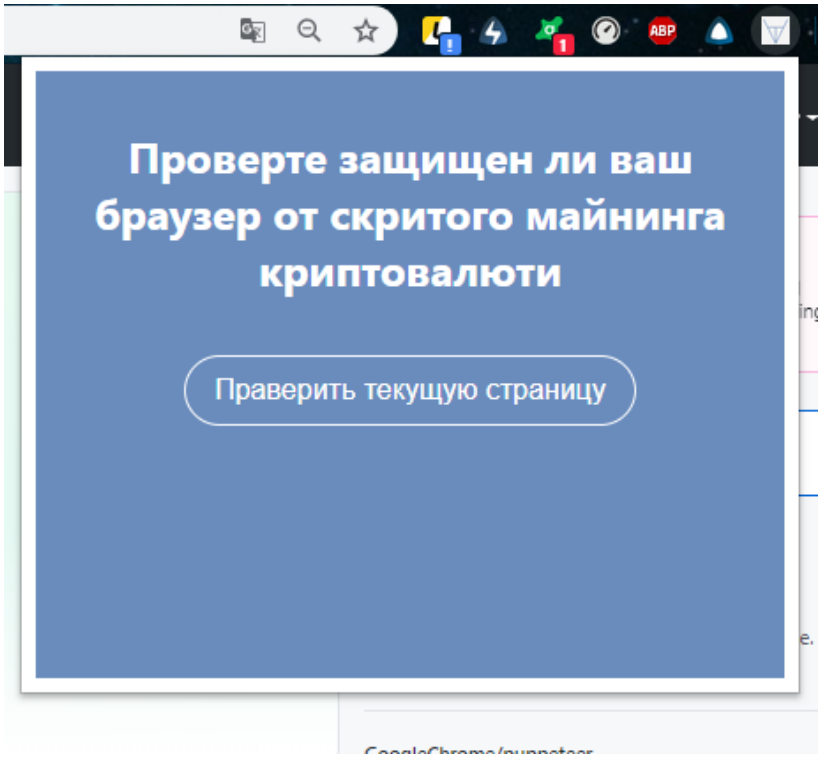

Рисунок 3.7 – Вигляд розширення після завершення розробки першої частини функціоналу

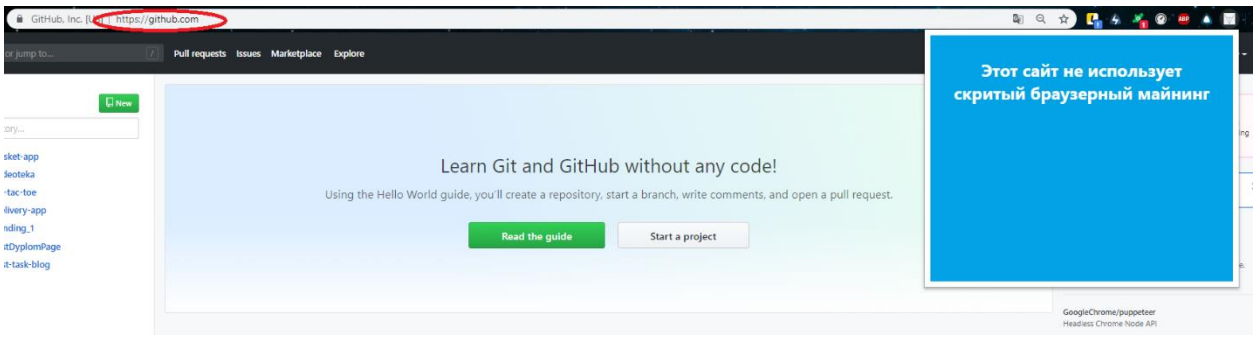

Рисунок 3.8 – Реакція розширення на сторінку github

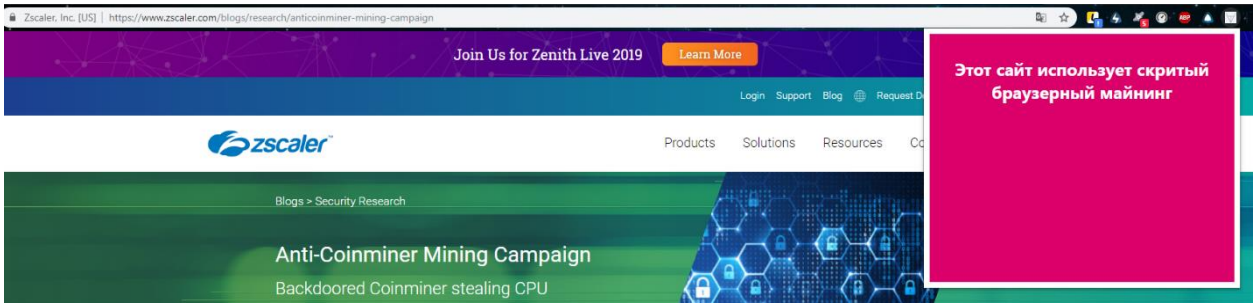

Рисунок 3.9 – Реакція розширення на сторінку, адреса якої входить до чорного списку

Для другого етапу перевірки перевірки я створив сторінку із входженням у код потенційно небезпечної бібліотеки та залив її на хост github.com(рис. 3.10). При перевірці сторінки розширення знаходить потенційно небезпечну бібліотеку і відповідно реагує(рис. 3.11).

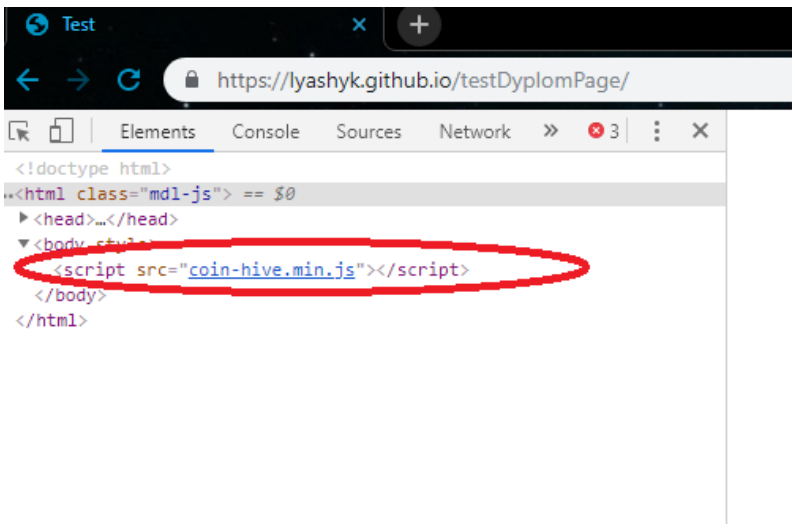

Рис 3.10 – Сторінка з бібліотекою, що забезпечує криптоджекінг

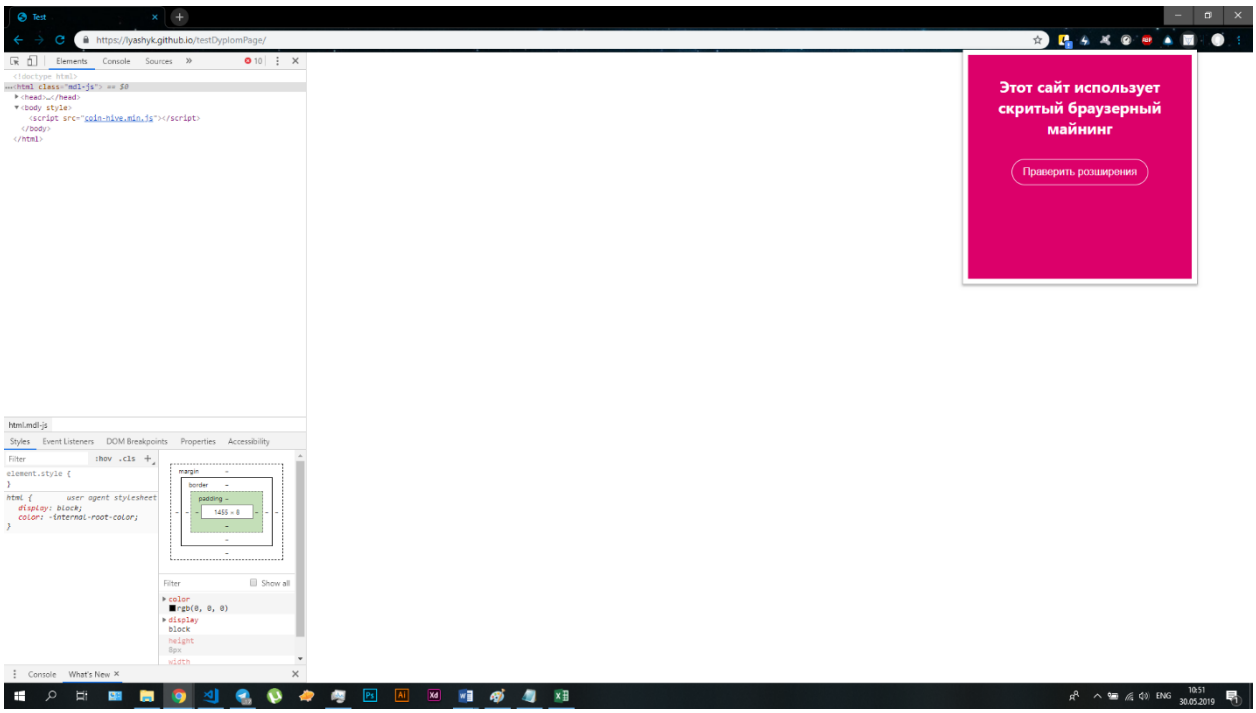

Рис 3.11 – Реакція розширення на заражену криптоджекінгом сторінку

3.6 Розробка функціоналу для пошуку криптоджекінгу у розширеннях Google Chrome

Іншою функцією розширення є перевірка розширень на предмет прихованого криптоджекінгу. Саме для цього ми у файлі-маніфесті вказали таке право, як manegament. Завдяки методу об’єкта chrome.manegament.getAll отримуємо доступ до списку розширень які встановлені у браузері. Потім імпортуємо js-файл, що містить список розширень, які були помічені за прихованим майнінгом криптовалюти(рис 3.12). Далі за допомогою функції indexOf перевіряємо список розширень, що ми отримали із функції getAll. Якщо виявляється, що відповідне розширення входить до списку, наш плагін одразу реагує відповідним повідомленням.

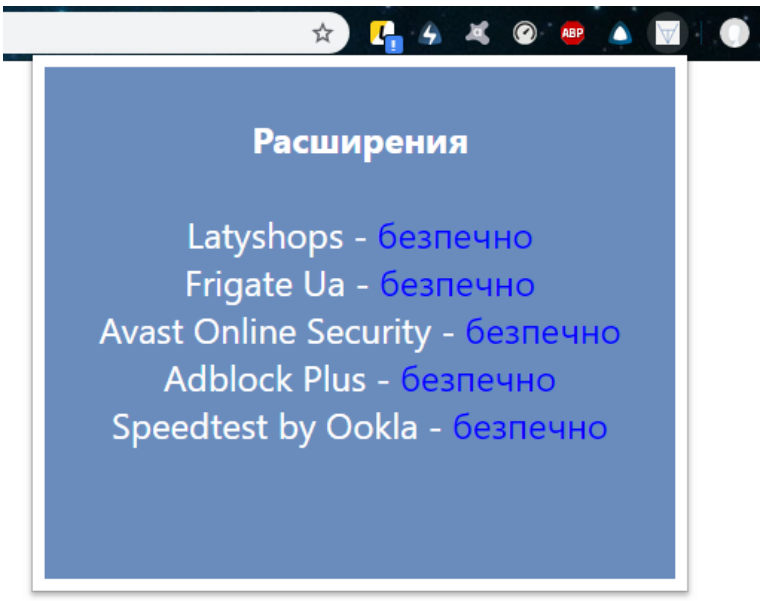

Рисунок 3.12 – Результат роботи розширення

3.7 Тестування розробленого функціоналу для пошуку криптоджекінгу у розширеннях Google Chrome

Для прикладу добавимо розширення під назвою LatyShops до чорного списку і побачимо що розширення це помічає і видає відповідний результат(рис 3.13).

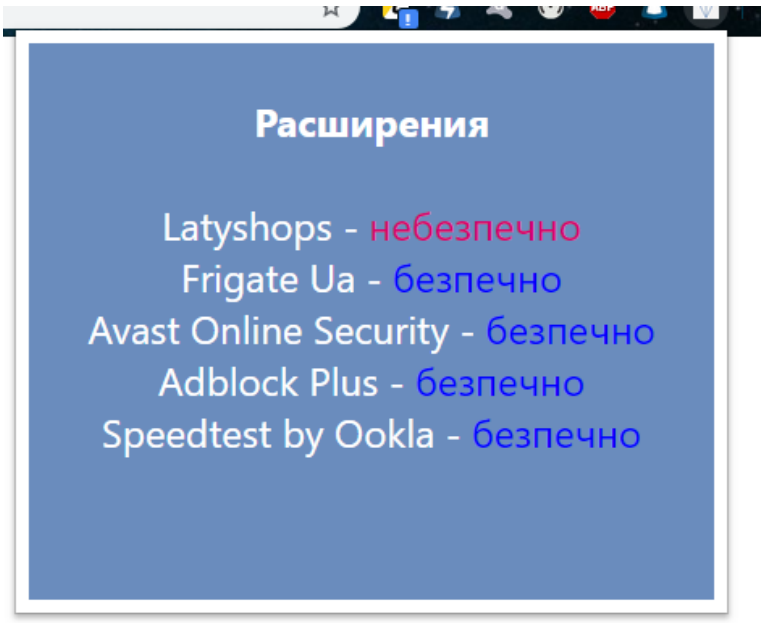

Рисунок 3.13 – Результат роботи розширення

3.8 Завантаження розширення на панель інстрементів Google Chrome

Отже розширення вже готове. Залишилося лише завантажити його до панелі інструментів Google Chrome. Отож вибираємо або завантажити до себе у браузер(рис 3.14) або запакувати проект у файл, для того щоб кожен з ким я поділився цим файлом - міг завантажити його собі у браузер(рис 3.15 - 3.17). Після того як ми завантажимо розширення собі у браузер ми зможемо його бачити у вікні разом з усіма іншими розшереннями(рис 3.18).

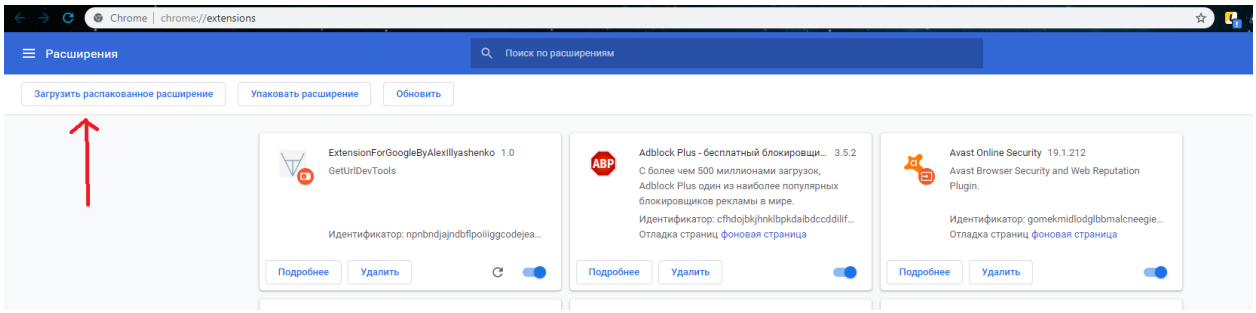

Рисунок 3.14 – Панель інструментів

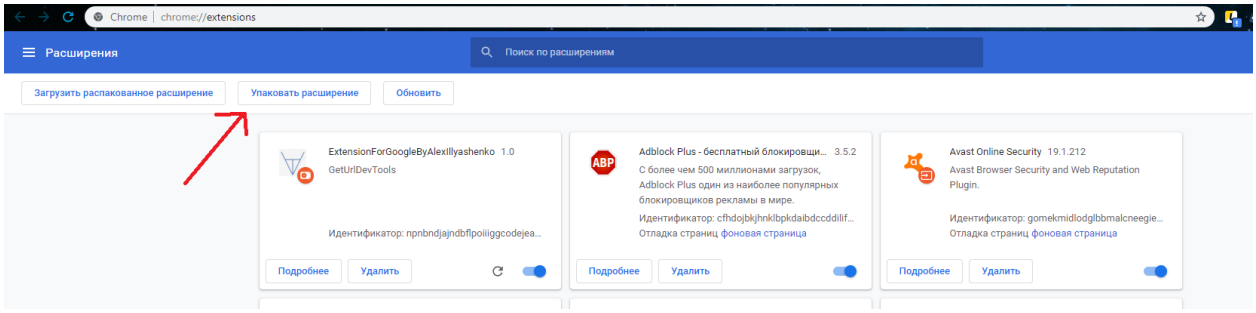

Рисунок 3.15 – Панель інструментів

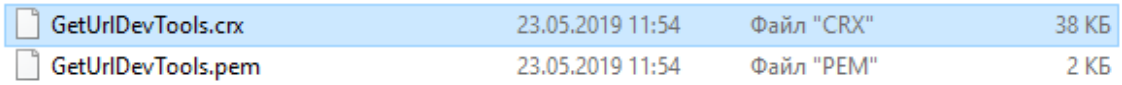

Рисунок 3.16 – Вже запаковане розширення, яким можна ділитись

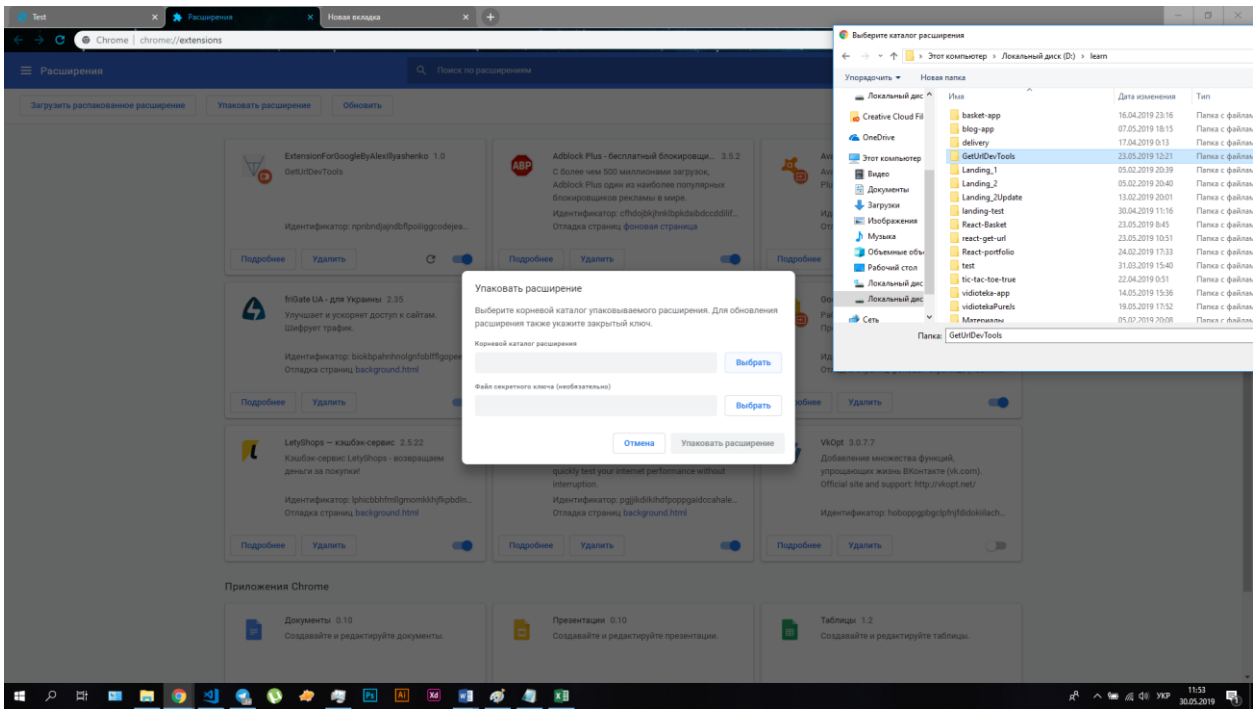

Рисунок 3.17 – Процесс запаковки розширення

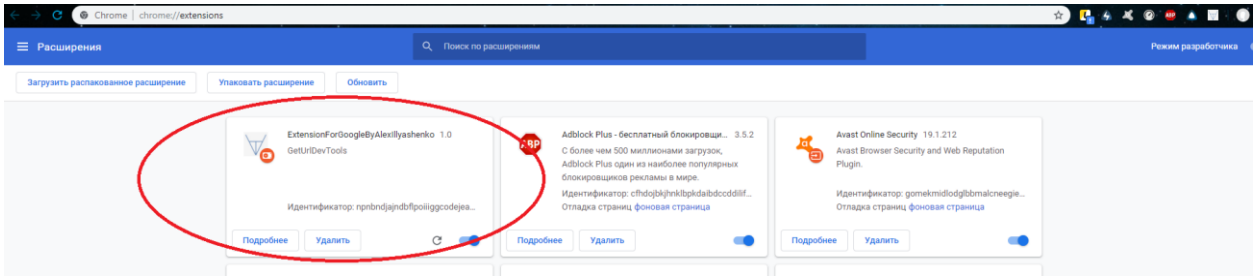

Рисунок 3.18 – Готове розширення на панелі Інструментів

Висновки до розділу 3

В данному розділі було запропоноване та реалізоване розширення для Google Chrome, що задовільняє деякі притензії, що було висуното у 2 розділі. А саме: написали розширення, яке не лише сповіщає користувача про наявний на сторінці криптоджекінг, залишаючи йому право вибирати між рекламою та погодженим видобутком криптомонети, а й можливість виявляти прихований майнінг криптомонет у самих розширеннях.

ВИСНОВКИ

У результаті даної роботи ми проаналізували ефективність методів виявлення прихованого браузерного криптоджекінгу та визначились із застосуванням кожного з них. Також з’ясували що у різних методів є різні призначення. Метод чорного списку та метод, який виконує пошук шкідливих бібліотек у коді програми застосовується беспосередньо для виявлення прихованого браузерного криптоджекінгу. Тертій метод, що відслідковує підозрілу поведінку зазвичай використовується для пошуку нових веб-сторінок, що заражені бібліотекою для прихованого майнінгу криптовалюти.

Проаналізувавши вже існуючі рішення було виявлено деякі проблеми. По перше усі прогамні засоби одразу блокують сайт із браузерним криптоджекінгом, не враховуючи те, що деякі веб-сторінки дають можливість на вибір: або погодитися на узгоджений криптоджекінг, або ж продивлюватися рекламні пропозиції. По друге на сьогоднішній день не було знайдено сервісів, які б давали можливість виявляти прихований криптоджекінг у розширеннях браузера.

Проаналізувавши проблеми, було розроблено розширення, яке рекомендується використовувати разом з іншими рішеннями, для усунення вищезгаданих проблем.

Перелік джерел посилань

1. Цифрова валюта. Визначення цифрової валюти. [Електронний ресурс] // wikipedia.org. – 2018. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0_%D0%B2%D0%B0%D0%BB%D1%8E%D1%82%D0%B0
2. Криптовалюта. Криптовалюта. [Електронний ресурс] // wikipedia.org. – 2019. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D0%B0%D0%BB%D1%8E%D1%82%D0%B0
3. Як не стати жертвою прихованого майнінгу. Визначення криптоджекінгу. [Електронний ресурс] // http://bitcoin-crypto-portal.com – 2018. – Режим доступу до ресурсу: http://bitcoin-crypto-portal.com/yak-ne-stati-zhertvoyu-prixovanogo-majningu/
4. Односторінковий застосунок. Визначення односторінкового застосунку. [Електронний ресурс] // wikipedia.org. – 2018. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5
5. Дерево DOM. Визначення DOM. [Електронний ресурс] // https://learn.javascript.ru – 2019. – Режим доступу до ресурсу: https://learn.javascript.ru/dom-nodes
6. React. Визначення react. [Електронний ресурс] // wikipedia.org. – 2019. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/React
7. AJAX. Визначення AJAX. [Електронний ресурс] // wikipedia.org. – 2019. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/AJAX

8. Брут-форс. Визначення брут-форсу. [Електронний ресурс] // www.securitylab.ru – 2005. – Режим доступу до ресурсу: https://www.securitylab.ru/news/tags/%E1%F0%F3%F2-%F4%EE%F0%F1/
9. Біткойн. Визначення біткойну. [Електронний ресурс] // wikipedia.org – 2019. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Bitcoin
10. Піца за 10 мільйонів. Перша покупка за біткойн [Електронний ресурс] // tsn.ua – 2017. – Режим доступу до ресурсу: https://ru.tsn.ua/groshi/picca-za-10-millionov-kak-podorozhal-bitkoin-na-primere-fastfuda-864407.html
11. Ботнет. Визначення Ботнету [Електронний ресурс] // wikipedia.org – 2019. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%91%D0%BE%D1%82%D0%BD%D0%B5%D1%82
12. Прихований майнінг криптовалют [Електронний ресурс] // cryptonet.biz – 2018. – Режим доступу до ресурсу: https://cryptonet.biz/ru/skrytyj-majning-kriptovalyut-kak-ustroen-zarabotok-i-imeet-li-perspektivy-na-rynke/
13. Shayan Eskandari, Andreas Leoutsarakos, Troy Mursch, Jeremy Clark. A First Look at Browser-Based Cryptojacking. - 2018. – c.2
14. The Pirate Bay снова майнит криптовалюту через браузеры посетителей [Електронний ресурс] // xakep.ru – 2017. – Режим доступу до ресурсу: https://xakep.ru/2017/10/12/tpb-mine-again/
15. Браузерный майнер Coinhive обнаружен в рекламе, размещенной на YouTube [Електронний ресурс] // xakep.ru – 2018. – Режим доступу до ресурсу: https://xakep.ru/2018/01/29/coinhive-on-youtube/
16. Браузерные майнеры научились использовать Google Tag Manager [Електронний ресурс] // xakep.ru – 2017. – Режим доступу до ресурсу: https://xakep.ru/2017/11/23/google-tag-manager-mining/
17. Зафиксированы массовые брутфорс-атаки на WordPress. [Електронний ресурс] // xakep.ru – 2017. – Режим доступу до ресурсу: https://xakep.ru/2017/12/21/wordpress-under-attack/

18. Geng Hong, Lei Zhang, Min Yan. How You Get Shotinthe Back: A Systematical Study about Cryptojacking in the Real World. - 2018. – c.5
19. Array.prototype.indexOf() [Електронний ресурс] // developer.mozilla.org – 2019. – Режим доступу до ресурсу: https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array/indexOf
20. JQuery [Електронний ресурс] // wikipedia.org – 2019. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/JQuery

ДОДАТОК А

Код файлу-маніфесту

```
{
  "manifest_version": 2,

  "name": "ExtensionForGoogleByAlexIllyashenko",
  "description": "GetUrlDevTools",
  "version": "1.0",

  "browser_action": {
    "default_popup": "popup.html"
  },

  "icons": {
    "16": "img.png",
    "48": "img.png",
    "120": "img.png"
  },

  "permissions": ["activeTab", "manegmant"]
}
```

ДОДАТОК В

Код html-файлу, що відповідає за динамічну побудову та відображення DOM-дерева

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

ДОДАТОК С

Код Js-файлу для вземодії html-файлу та jsx-файлу

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './containerApp';

ReactDOM.render(<App />, document.getElementById('root'));
```

ДОДАТОК D

Код Jsx-файлу, що є основним у проекті

```
import React, { Component } from 'react';
import chrome from 'react-chrome';
import cn from 'classnames';

import Data from './blacklist';
import $ from './jquery-3.4.1.min.js';
import dengerLib from './denjer-library';
import dE from './dangerExpansion';

const TITLE = {
  DEFAULT: 'Проверте защищен ли ваш браузер от скритого майнинга криптовалюты',
  NORMAL: 'Этот сайт не использует скритый браузерный майнинг',
  DENGER: 'Этот сайт использует скритый браузерный майнинг',
  NOTHIMG: ''
};

const RES = {
  TRUE: 'безпечно',
  FALSE: 'небезпечно'
};

export default class ContainerApp extends Component {
  state = {
    isCheckGood: false,
    isCheckBad: false,
    titleContent: TITLE.DEFAULT,
    battonPage: false,
    battonExpansion: false,
    contentExpansionRes: '',
    contentExpansion: ''
  };
  render() {
    return (
      <div
        className={
          (cn('container'),
          {
            'container--normal': this.props.isCheckGood,
            'container--denger': this.props.isCheckBad
          })
        }
      >
        <h6 className="title">{this.props.titleContent}</h6>
        <button
          className={
            (cn('buttonPage button'),
```

```
          {
            'buttonPage--none': this.props.battonPage
          })
        }
        onClick={this.onClickButtonPageFunction}
      >
        Праверить текущую страницу
      </button>
      <button
        className={
          (cn('buttonExpansion button'),
          {
            'buttonExpansion--none': this.props.battonExpansion
          })
        }
        onClick={this.onClickButtonExpansionFunction}
      >
        Праверить текущую страницу
      </button>
      <div className="contenFlexWrapper">
        <div className="contentExpansion">{this.props.contentExpansion}</div>
        <div className="contentExpansion">
          {this.props.contentExpansionRes}
        </div>
      </div>
    </div>
  );
}

onClickButtonPageFunction = () => {
  this.setState({
    battonPage: true
  });
  this.checkBlackList();
};

onClickButtonExpansionFunction = () => {
  this.setState({
    battonExpansion: true
  });
  this.checkBlackListExpansion();
};

checkBlackList() {
  chrome.tabs.getSelected(null, this.assignmentUrlFunction);
}

checkBlackListExpansion() {
  const expansionList = chrome.manegmant.getAll();
  const resultList = expansionList.map(item => {
    return dE.indexOf(item) >= 0;
```

```
    });
    const expansionListContent = expansionList.map(item => {
      return <div className="flexChild">item</div>;
    });
    const expansionListResContent = resultList.map(item => {
      return <div className="flexChild">{RES.item}</div>;
    });
    this.setState({
      contentExpansionRes: expansionListResContent,
      contentExpansion: expansionListContent
    });
  }

  assignmentUrlFunction(tab) {
    const b = $.ajax(tab.url);

    b.done(function(d) {
      const div = document.getElementsByClassName('div')[0];
      const resultIndexOfScript = dengerLib.map(item => {
        return d.indexOf(item) >= 0;
      });
      const a = !!(resultIndexOfScript.indexOf(true) >= 0);

      div.textContent = a;
    });
    const res = document.getElementsByClassName('div')[0].textContent;
    if (Data.indexOf(tab.url) >= 0 && res) {
      this.setState({
        isCheckGood: true,
        isCheckBad: false,
        titleContent: TITLE.NORMAL
      });
    } else {
      this.setState({
        isCheckGood: false,
        isCheckBad: true,
        titleContent: TITLE.DENGER
      });
    }
  }
}
```

ДОДАТОК Е

Файл стилів для основного jsx-файлу

```
.container {
  background-color: #6a8cbd;
  margin: 0;
  padding: 32px;
  width: 360px;
  height: 280px;
  color: #fff;
  text-align: center;
  font-size: 24px;
}

.container--normal {
  background-color: #01a3e6;
}

.container--denger {
  background-color: #dc006a;
}

.title {
  margin: 0;
  font-size: 24px;
  margin-bottom: 32px;
}

.button {
  color: #fff;
  display: inline-block;
  padding: 8px 16px;
  margin: 0;
  border: solid 1px #fff;
  border-radius: 20px;
  line-height: 22px;
  font-size: 16px;
  background-color: transparent;
  cursor: pointer;
  outline: none;
  transition: all 0.2s ease-out;
  appearance: none;
  margin-bottom: 16px;
}

.buttonPage--none {
  display: none;
}
```

```
.buttonExpansion--none {
  display: none;
}

.button:hover {
  background-color: rgba(255, 255, 255, 0.1);
}

.button:active {
  background-color: rgba(255, 255, 255, 0.2);
}

.contenFlexWrapper {
  display: flex;
  margin: -16px;
}

.contentExpansion {
  padding: 16px;
  display: flex;
  flex-direction: column;
  margin: -8px;
}

.flexChild {
  padding: 8px;
}
```