Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка та впровадження web-сервісу захищеного інтернет-серфінгу»

Студента 2 курсу, 5м групи, спеціальності 121 «Інженерія програмного забезпечення», спеціалізації «Інженерія програмного забезпечення»

_____ підпис студента

Залевського Богдана Паловича

Науковий керівник кандидат технічних наук, професор

_____ підпис керівника

Пашорін Валерій Іванович

Гарант освітньої програми доктор технічних наук, професор

_____ підпис керівника

Криворучко Олена Володимирівна

КИЇВ – 2019

АНОТАЦІЯ

Залевський Б.П. Розробка та впровадження web-сервісу захищеного інтернет-серфінгу

У випускній кваліфікаційній роботі розглянуто розробку web-сервісу, його компонентів та функціональності, а також використання PHP, та технології Apache HTTP Server для побудови web-додатків. Також були розглянуті існуючі загрози та програмні методи захисту користувачів в інтернету.

Ключові слова: Розробка програмного забезпечення, програмне забезпечення, PHP, Apache, web-сервіс, проксі.

ANNOTATION

Zalevsky B.P. Development and implementation of a secure web surfing web service

In the final work examines the development of a web service, its components and functionality, as well as the use of PHP and Apache HTTP Server technology to build web applications. Existing threats and software methods for protecting users on the Internet were also considered.

Keywords: Software development, software, PHP, Apache, web service, proxy.

ТЕХНІЧНЕ ЗАВДАННЯ WEB-СЕРВІСУ ЗАХИЩЕНОГО ІНТЕРНЕТ-СЕРФІНГУ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Web-сервіс захищеного інтернет-серфінгу

Галузь застосування: Інформаційні технології, кібербезпека

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на Випускну кваліфікаційну роботу, затверджене кафедрою програмної інженерії та кібербезпеки.

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Проект має являти собою захищений web-сервіс що надає можливість переглядати інтернет-сторінки через внутрішній інтерфейс за допомогою проксі з’єднання. Призначенням такої системи є організація захищеного інтернет-серфінгу користувачам, захист від небажаного відслідковування та зберігання анонімної особистості.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Сервіс має бути доступний як онлайн, так і для локального запуску та тестування. Для цього усі необхідні файли потрібно буде завантажити на хостинг. Вибір хостингу має опиратися на різноманітність серверів, високий час роботи та відмовостійкість.

При відкритті сторінки, її змістом має бути поле для введення адреси призначення, та реалізована можливість вибору одно із декількох серверів.

При переході за необхідною адресою, користуючись сервісом, користувачеві має стати доступна сторінка з посилання, але бути відображеною не зі сторони клієнта а зі сторони сервера, тобто створюючи проксі з'єднання та відображаючи захищенний контент. Web-сторінка що переглядається має містити додаткову форму з функцією зміни адреси чи повернення на стартову сторінку. Також повинно бути реалізовано переключення між різними серверами. Сервери мають однакові за оформленням сайти, з аналогічним функціоналом але з різними геолокаційними характеристиками та IP-адресами.

В якості мови програмування було обрано PHP версії 7.3.

Локальне налагоджування та тестування за допомогою Apache HTTP Server.

Передбачається що сервіс буде повністю функціональний та сумісний з версіями таких інтренет браузерами, як Google Chrome, Opera, Mozila Firefox. Сервіс та має бути сумісний і для комфортного перегляду на екранах невеликого розміру сучасних смартфонів. Первинна розробка та тестування відбуватиметься браузером Opera.

Web-сервіс повинен мати наступну функціональність:

- Відображати зрозумілий інтерфейс;
- Приймати та опрацьовувати уведені посилання користувача;
- Перенаправляти запит на проксі-сервер;
- Відобразити сторінку посилання через посередництво проксі серверу;
- Відображати увесь зміст заданої сторінки;
- Шифрувати URL адресу;
- Можливість зберігати сторінку у закладках браузеру.

5. ВИМОГИ ДО ПРОГРАНМОГО ЗАБЕЗПЕЧЕННЯ

Технології що будуть застосовуватися при побудові:

- Утиліта Apache HTTP сервер;
- Мова програмування PHP 7.3;
- Текстовий редактор Sublime Text 3 для кодування;
- Технологія хосту сайтів та серверів 000webhost.

6. ВИМОГИ ДО ПРОЕКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання роботи повинна бути розроблена наступна документація:

1) Технічне завдання;
2) Програма та методика користувача;
3) Керівництво користувача.

7. ЕТАПИ ПРОЕКТУВАННЯ

1) Вивчення літератури за тематикою роботи;
2) Розробка та узгодження технічного завдання;
3) Розробка моделі функціонування;
4) Розробка структури web-сервісу;
5) Програмна реалізація web-сервісу;
6) Тестування web-сервісу;
7) Підготовка матеріалів текстової частини роботи;
8) Підготовка матеріалів графічної частини роботи.

Київський національний торговельно-економічний університет

Факультет ФОАІС Кафедра «Програмної інженерії та кібербезпеки»

Спеціальність 121 «Інженерія програмного забезпечення»

Спеціалізація «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри

_____Криворучко О.В.

«___» _____2019 р.

Завдання
на випускну кваліфікаційну роботу студентові

Залевському Богдану Павловичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи «Розробка та впровадження web-сервісу захищеного інтернет-серфінгу»

Затверджена наказом ректора від «07» *листопада* 2018 р. № 4183

2. Строк здачі студентом закінченої роботи (проекту)_____

3. Цільова установка та вихідні дані до роботи (проекту)

Мета роботи – розробка та впровадження web-сервісу захищеного інтернет-серфінгу».

Об'єкт дослідження - *всесвітня мережа інтернет, проблематика захисту анонімності користувачів, загальні рекомендації організації захищеного інтернет з'єднання.*

Предмет дослідження – програмні засоби забезпечення додаткового захисту користувача, приватні віртуальні мережі, використання проксі-серверів, функціональність такого програмного забезпечення, web-орієнтовані мови та методи програмування.

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1.ФУНКЦІЇ ТА СКЛАДОВІ БЕЗПЕЧНОЇ РОБОТИ В МЕРЕЖАХ

1.1. Актуальні загрози персональним даним та анонімності користувачів при роботі в відкритих мережах

1.2. Концепція мережевої гігієни та безпечної роботи в інтернет мережіі

1.3. Висновки до розділу 1

РОЗДІЛ 2. ІНСТРУМЕНТАЛЬНІ ЗАСОБИ МЕРЕЖЕВОЇ БЕЗПЕКИ

2.1. Загальна характеристика та класифікація засобів захисту

2.2. Аналіз ефективності існучих засобів захисту безпеки користувачів

2.3. Розробка моделі захищенного інтернет-серфінгу

2.4. Висновки до розділу 2

РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ WEB-СЕРВІСУ ЗАХИЩЕНОГО ІНТЕРНЕТ-СЕРФІНГУ

3.1. Вибір програмних засобів та мови програмування

3.2. Загальна характеристика та спосіб функціонування системи

3.3. Реалізація та впровадження системи

3.4. Висновки до розділу 3.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	*Вибір теми випускної кваліфікаційної роботи*	*25.09.2018*	
2.	*Розробка та затвердження завдання на проект магістра (Наказ №185 від 07.11.18)*	*07.11.2018*	
3.	*Вступ та перелік літературних джерел*	*28.02.2019*	
4.	*Розробка технічного завдання*	*22.03.2019*	
5.	*Розділ 1. Функції та складові безпечної роботи в мережах*	*18.04.2019*	
6.	*Розділ 2. Інструментальні засоби мережевої безпеки*	*24.05.2019*	
7.	*Розділ 3. Розробка та впровадження web-сервісу захищеного інтернет-серфінгу*	*21.06.2019*	
8.	*Розробка програми та методики тестування*	*18.10.2019*	
9.	*Написання наукової статті*	*27.09.2019*	
10.	*Керівництво користувача*	*23.10.2019*	
11.	*Висновки та пропозиції*	*01.11.2019*	
12.	*Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)*	*13.11.2019*	
13.	*Підготовка автореферату та презентації доповіді*	*14.11.2019*	
14.	*Попередній захист випускної кваліфікаційної роботи*	*14.11.2019 – 15.11.2019*	
15.	*Здача зброшурованої випускної кваліфікаційної роботи*	*25.11.2019*	
16.	*Зовнішнє рецензування випускної кваліфікаційної роботи*	*26.11.2019*	
17.	*Підготовка до публічного захисту випускної кваліфікаційної роботи*	*02.12.2019-04.12.2019*	

7. Дата видачі завдання **«26» вересня 2018 р.**
8. Науковий керівник випускної кваліфікаційної роботи

_____

(прізвище, ініціали, підпис)

9. Гарант освітньої програми _____

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент_____

(прізвище, ініціали, підпис)

11. Відгук керівника випускної кваліфікаційної роботи

____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____

Науковий керівник випускної кваліфікаційної роботи___________________

(підпис, дата)

Відмітка про попередній захист ____

(ПІБ, підпис, дата)

12. Висновок про випускну кваліфікаційну роботу

Випускна кваліфікаційна робота студента ______________________________

(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми ____

(прізвище, ініціали, підпис)

Завідувач кафедри ____

(підпис, прізвище, ініціали)

ЗМІСТ

					КНТЕУ 121 - 05-12.МР		
Зм.	*Аркуш*	*№ докум.*	*Підпис*	*Дата*	*Розробка та впровадження web-сервісу захищеного інтернет-сервісу*	*Стадія*	*Аркуш*	*Аркушів*
Зав. каф.		Криворучко О.В.				*3*	*2*	*68*
Керівник		Пашорін В.І.						
Гарант		Криворучко О.В.			*Зміст*	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		
Розробив		Залевський Б.П.						

ВСТУП

Становлення інформаційного суспільства пов'язано з широким розповсюдженням комп'ютерної техніки, мобільних пристроїв, побудовою інтернету, таким як ми бачимо його сьогодні, та підключенням до мережі все більшого числа користувачів. З тих пір, як комп’ютери з’явилися майже у кожному домі, а кількість активних смартфонів перевалила за 4,3 мільярди, велика частина інформації зберігається у цифровому вигляді. Ці досягнення змінили спосіб життя суспільства, виставивши на перший план завдання з обробки, зберігання та найголовніше обміну цими даними.

Перехід до застосуваня електронних носіїв інформації призвів до проблем пов’язаних з конфіденційною передачею інформації. Надійний захист персональних даних є важливою задачею для вирішення сучасними фахівцями. Навіть в великих, популярних web-сервісах, сховищ даних чи соціальних мережах залишається загроза небажаного відслідковування, перехоплення трафіку чи компрометації персональних даних.

Актуальність обраної теми полягає в щоденному користування мережі інтернет понад мільярдом користувачів, що допомагає як в покращенні зв’язку і передачі даних, так і створенні проблем захисту персональних даних, конфіденційної інформації та анонімності в цілому.

Метою даної випускної кваліфікаційної роботи є розробка, тестування та впровадження web-сервісу захищеного інтернет серфігну. Згідно з

поставленою метою, основними завданнями випускної квалфікаційної роботи є:

- Здійснити аналіз актуальних загроз персональних даних та анонімності;
- Розробити загальні рекомендації мережевої гігієни;
- Аналіз та опис існуючих програмних засобів захисту в мережі;
- Проектування моделі захищеного інтернет-серфінгу;
- Розробка та впровадження web-сервісу захищеного інтернет серфінгу;
- Тестування розробки.

Об’єктом дослідження є всесвітня мережа інтернет, проблематика захисту анонімності користувачів, загальні рекомендації організації захищеного інтернет з’єднання.

Предметом дослідження є програмні засоби забезпечення додаткового захисту користувача, приватні віртуальні мережі, використання проксі-серверів, функціональність такого програного забезпечення, його структура та складові, web-орієнтовані мови програмування та методи проектування.

Наукова новизна полягає в розробці web-сервісу на базі доповнених, уточнених даних актуальних загроз та існуючого пз, з використанням сучасних мов програмування та технічної документації.

Методи дослідження: Інформаційні бази, авторські роботи, наукові журнали та періодичні видання, посібники з програмування та web-програмування, енциклопедичні дані, інтернет ресурси пов'язані з програмуванням.

РОЗДІЛ 1
ФУНКЦІЇ ТА СКЛАДОВІ БЕЗПЕЧНОЇ РОБОТИ В МЕРЕЖАХ

1.1. Актуальні загрози персональним даним та анонімності користувачів при роботі в відкритих мережах

До персональних даних можна віднести будь-які відомості, за якими ідентифікується або може бути ідентифікована фізична особа, зокрема: прізвище, ім'я, адреса, телефони, паспортні дані, національність, освіта, дата і місце народження, місце проживання та перебування тощо. Така інформація про користувача є конфіденційною і може оброблятися в тому числі поширюватись тільки за його згодою, крім випадків, визначених законом. Політика безпеки багатьох ресурсів передбачає зберігання конфіденційної інформації та персональних даних користувача чи не найвищим пріоритетом, але часто ми можемо бачити її порушення через банальне недбальство або свідомі злочинні наміри.

Під актуальною загрозою безпеки персональних даних розуміється сукупність умов і факторів, що створюють актуальну небезпеку несанкціонованого, в тому числі випадкового, доступу до персональних даних при їх обробці в інформаційній системі, результатом якого, можуть стати знищення, зміна, блокування, копіювання, надання, поширення персональних даних, а також інші неправомірні дії.

Сучасною поширеною загрозою є витік персональної інформації із соціальних мереж, поштових сервісів чи просто баз даних. Прикладом такого витоку є витік персональних даних майже 50 млн. користувачів, мережі Instagram у травні 2019 року. Як виявилося база даних лежала у відкритому

доступі на Amazon web Service, і кожен охочий міг отримати доступ до конфіденційних даних [4]. Крім того, база постійно поповнювалася свіжими особистими даними користувачів які і не підозрювали про витік інформації. В даному прикладі наведено несанкціонований, випадковий доступ до даних пов'язаний з недбальством а не зі злочинним умислом. В той же час компанія Facebook призупинила роботу десятків тисяч додатків приблизно 400 розробників. Це сталося після масивних витоків інформації користувачів коли додатки використовували персональні дані користувачів Facebook у неправомірних цілях.

Персональна інформація може неправомірно розповсюджуватися і співробітниками компаній що мають до неї доступ. Так у кінці 2018 року працівники Київського управління Департаменту кіберполіції Національної поліції України викрили діяльність трьох громадян України, які організували схему збуту інформації з обмеженим доступом та персональних даних громадян, за допомогою мережі інтернет. Інформацію стосовно громадян зловмисники отримували з баз даних однієї з кредитних спілок та бази ДФС, при цьому доступ до цих баз надавали співробітники організацій, які входили до складу злочинної групи [6]. Багато подібних даних продаються на відкритих та закритих форумах, Telegram каналах чи даркнет платформах саме персоналом що має до них доступ, або довіреними особами. Іноді це стосується навіть державних даних систем, та не дивлячись на законодавство України, інсайдерами з доступом до такої інформації можуть бути службовці правоохоронних та загальнодержавних органів, міністерств, організацій тощо.

Захист анонімності відрізняється від захисту персональних даних. В той час як персональні данні можуть бути анонімними, та зберігатися на анонімних засадах згідно політики захисту персональних даних, анонімність як поняття передбачає часткову або повну відсутність такої інформації про певного користувача, чи групи осіб взагалі. Багато хто сприймає анонімність

як право, особливо в інтернет-спілкуванні. Часткове право на анонімність до певної міри захищене законом у різних юрисдикціях.

Загроза анонімності вбачається не у витоці даних чи їх передачі третім особам, а деанонімізації – розкриття певних персональних даних особи, чи групи осіб, яка бажала залишатися повністю або частково анонімною. У відкритих мережах найчастіше така загроза реалізується визначенням ір-адреси пристрою, яким користувалася особа. Ще один спосіб деанонімізаці є пошук за ключовими факторами того сліду що анонімна особа залишила на певному сайті чи web-ресурсі. Наприклад користувач використав один і той же нікнейм до різних web-сайтів, один із який надає певну персональну інформацію реальної особи, чи було оприлюднено певний файл (найчастіше фотографію) що мають приховані метадані чи інші деанонімізуючі атрибути.

Загалом сучасні зарози оприлюднення інформації що містять персональні дані можна класифікувати за способом та цілями виконання:

- Добровільна – користувач свідомо повідомляє особисту інформацію про себе чи певну групу осіб;
- Випадкова – користувач чи довірена особа що має доступ до персональних даних (компанія, корпорація) за певних обставин випадково розкриває особисту інформацію про себе чи певну групу осіб;
- Зловмисна – розголошення особистих даних особи чи групи осіб з недобросовісними намірами. Цілями можуть бути як фінансове збагачення, так і моральне цькування чи вандалізм, в тому числі с особистих, політичних, релігійних міркувань тощо.

Підставні web-сайти та додатки можуть використовуватися для сбору таких даних. Користувач заповнює свої реальні дані на підставних вер-ресурсах чи вспливаючих вікнах, в той час як зловмисник (таких ще називають «скамери») перехоплює ці дані, і вже на офіційному ресурсі

використовує їх для викрадення особистої інформації, документів, фото та привласнення усього акаунту, особливо якщо він має цінність у вигляді платного сервісу чи фінансової установи.

Щоб результатом дій на сайті не став витік особистих даних або спам-атака на пошту, рекомендовано користуватися ресурсами, мережева адреса яких починається з HTTPS. Це означає, що з'єднання між користувачем і сервером здійснюється у зашифрованому - інформацію не вийде перехопити. Сучасні браузери нагадують про ризик нешифрованих з'єднання. Якщо при перевірці сертифіката дані не збігаються, на екрані з'являється попередження.

Особливо небезпечно так втратити приватний фінансовий чи банківський акаунт, позбутися реальних коштів, криптовалюти чи певних нематеріальних активів.

Проблемою мережі можна виділи саме підключення, навіть коли ми використовуємо захищений поштовий клієнт, сторінку для автентифікації в соціальній мережі чи банквского акаунту. Мова йде про перехоплення трафіку, або «сніфінг». На сьогодні роблему видокремлено більше до безпровідних wifi мереж. При використані домашнього інтернету це зазвичай не загрожує, а ось трафік публічної wifi мережі можна перехопити навіть без спеціальних технчних засобів, маючи лише смартфон чи ноутбук зі встановленим спеціальним програмним забезпеченням (tcpdump, Wireshark, York чи Airodump-ng). Найчастіше відкритий wi-fi встановлюсь у різноманітних закладах ресторанного господарства (в тому числі кафе), торгових центрах, громадському транспорті, бібліотеках, навчальних закладах чи навіть банках для зручності відвідувачів чи доступу онлайн бібліотек. Через свою відкритість такі мережі часто стають інструментом розповсюдження шкідливого пз та перехопленням даних користувачів. Деякі мережі вимагають додаткової реєстрації окремого користувача, чи додаткового паролю який хоча і нескладно отримати вільно, та все ж запобігає хаотичним підключенням пристроїв. Але часто цим нехтують,

залишаючи мережу відкритою для кожного, в тому числі зловмисника. Крім того, деякі оператори мереж wi-fi навмисно вводять рекламу в web-трафік, що може привести до небажаного відстеження.

Як саме може бути реалізована загроза в подібних інтернет мережах, можна переглянути на прикладі набору Aircrack-ng [7]. Це існуючий спеціальний набір програм що містить утиліти виявлення та перехоплення бездротових мереж, ключів шифрування, перевірки схильності атак (пентест) тощо. Як і багато подібних утиліт кібербезпеки його можна використовувати і зловмисно. В незахищеній мережі утиліти даного набору дозволяють сканувати мережу на наявні пристрої, та відмічати MAC-адреси пристроїв що були підключені (Рис. 1.1). Через деякий час зловмиснику достатньо змінити MAC-адресу свого WI-FI адаптеру та підключитися до вже знайомої мережі під виглядом абсолютно іншого пристрою, з доступом до його інтернет даних.

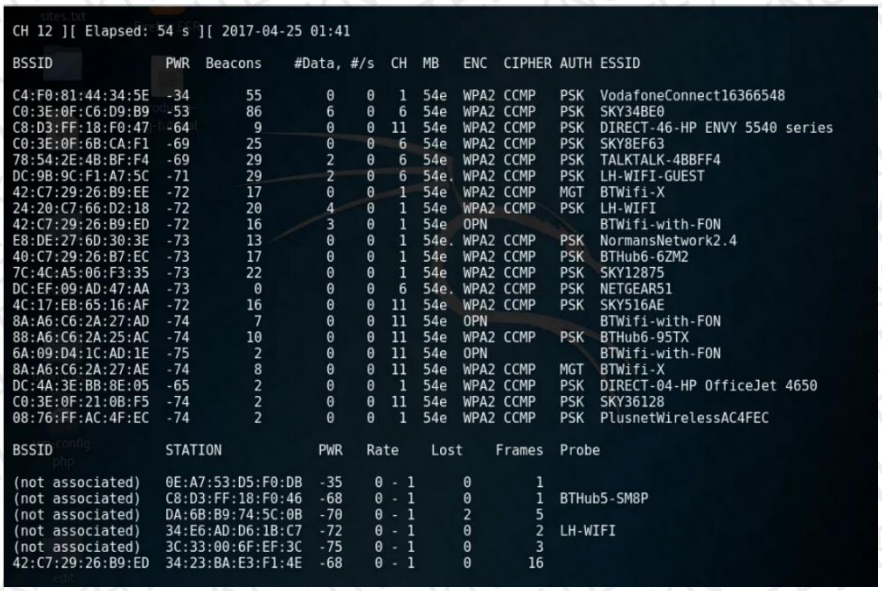


Рис. 1.1. Режим моніторингу підключень утиліти Airodump-ng.

1.2. Концепція мережевої гігієни та безпечної роботи в інтернет мережі

На основі існуючих нюансів роботи у комп'ютерних мережах, та оброки даних користувачів важливо сформувати певні правила безпечного користування такими мережами. Концепція мережевої гігієни являє собою рекомендації та загальні правила безпечного поводження в мережі інтернет, з метою запобігання витоку персональної інформації, втрати даних та нівелювання існуючих загроз для користувача. Концепція передбачає наступні правила безпечного поводження, але і цей список не є вичерпним та залежить від обережності користувача:

- Використання різних паролів суттєво зменшує ризик доступу до ваших персональних даних у разі витоку інформації. Так отримавши пароль від одного з акаунтів, зловмисник може підібрати його і до інших, в тому числі поштових сервісів. Періодична зміна паролів ще більше покращує такий захист, а для того щоб паролі не доводилося вигадувати, можна користуватися спеціальними генераторами. Для того, щоб паролі не доводилося щоразу вигадувати, можна користуватися спеціальними генераторами. Крім різноманітності, вони також створюють безпечні комбінації, що складаються з цифр, символів і літер різного регістра;
- Не варто повідомляти пару «логін/пароль» стороннім особам, навіть якщо вони назвалися представниками адміністрації ресурсу чи служби підтримки. Багато сервісів де користувач може вести особистий кабінет завжди попереджають про подібні випадки, та налогошують що співробітники ніколи не просять паролі своїх клієнтів;
- За можливості використовувати двофакторну автентифікацію;
- За потреби користуватися пошуковими системами що не відслідковують дії користувача, наприклад, DuckDuckGo.com не збирає дані про своїх відвідувачів, не зберігає IP-адреси і cookies;

- Не відкривати підозрілі посилання з надходжених листів, та не завантажувати надіслані файли. Завжди перевіряти адресата, та не довіряти підозрілим повідомленням;
- Навіть якщо незрозуміле посилання надійшло від знайомого адресата, потрібно переконатися що це саме він. Через знайомі облікові записи часто відбуваються зломи та зараження що ланцюгово передаються;
- Завжди перевіряти сайти та посилання на них. Перш ніж завантажити щось з незнайомого джерела, уводити паролі або довіряти представленій інформації варто перевірити сторінку. На рис. 1.2 зображено приклад фішингу за допомогою підробленої web-сторінки. Не дивлячись на сертифікат безпеки uk-ua.facebook.com э лише субдоменом beck.com, та є підробкою сторінки входу у Facebook ;

Рис 1.2. Приклад підробленої web-сторінки.

- У відкритих джерелах не бажано мати особистий телефон, використовувати робочі номери для робочих питань, або мати додатковий номер спеціально для інтернет сервісів та соціальних мереж. Навіть через працівників стільникових операторів можна отримати ваші анкетні дані, а далі і будь-яку іншу інформацію. До того ж це вберігає від зайвого спаму, СМС та дзвінків;
- Адреси небезпечно викладати в мережу через загрозу відстежування, пограбування чи вандалізму. В соціальних мережах рідко хто залишає таку інформацію, але її і досі можна отримати за допомогою фото, геолокації чи соціальної інженерії під час спілкування. Те ж стосується і інформацію про IP чи MAC-адреси;
- Зайвий раз не розлогошувати дані банківських рахунків чи кредитних карток. Навіть сьогодні при використанні двофакторної автентифікації на банківських рахунках є загроза стати жертвами кардерів, та додати

свою картку до списку «зламаних кредиток» що так активно продаються в даркнеті;

- Обов’язково утриматись від викладування особистих документів у загальний доступ. Паспорти і водійські посвідчення, отримані від необережних власників, користуються попитом на тіньових ресурсах. Вони дуже дешево коштують і застосовуються в різних схемах, починаючи від отримання кредитів і закінчуючи розміщенням на дошках оголошень, коли шахрай висилає чужий документ, щоб підтвердити свою особу;
- Обмежети користування публічними wi-fi точками. Якщо така потреба виникає не варто логінитись до акаунтів чи користуватися фінансово-банківськими системами;
- Домашні wi-fi мережі потрібно обов’язково спорядити власним паролем, як на саме підключення так і на адміністративну панель керування. Усі сучасні роутери постачаються з інструкцією, вимоги якої слід обов’язково дотримувати.

Безпечна робота в мережі інтернет не закінчується лише обережністю. Існують багато спеціальних засобів захисту на базі програмного забезпечення чи відповідних web-ресурсів.

Безпечне користування мережею не обмежується свідомими діями користувача. Для захисту як домашніх так і корпоративних мереж варто використовувати і спеціалізоване апаратне та програмне забезпечення, сервіси чи навіть надбудовані комп’ютерні мережі. Безпека інформаційної мережі включає захист, програмного забезпечення, обладнання, даних та користувачів. Безпечне користування мережами досягається за допомогою багаторівневого захисту, до якого входить з’єднання, захист роутера, робочих станцій та окремих пристроїв. Для виявлення шкідливих програм, та запобігання їх негативного ефекту використовується різноманітне антивірусне забезпечення, мережеві екрани або системи запобігання

вторгнень (IPS). Зв'язок між двома комп'ютерами з використанням мережі можна зашифровувати, щоб зберегти конфіденційність.

Захист самої мережі та захист даних користувача можуть мати схожі так і окремо різні функції. Захист мережі в більшості стосується захисту самої мережі, її відмовостійкість та недопущення негативного впливу, в той час як захист даних користувача є більш локальним питанням, з проблемами конфіденційності чи захисту персональних даних. Це є актуальним навіть при користуванні web-сервісах в захищених мережах. Методи захисту таких даних включають захист саме особистого з'єднання з використанням проксі-серверів, віртуальні приватні мережі або надбудови над існуючими мережами, хоча можуть і використовуватися великими корпоративними мережами як додатковий шар захисту над стандартними методами. За функціоналом та методом використання такі рішення безпеки відрізняються.

Високий захист мережевого з'єднання користувача передбачає і ряд недоліків. При під'єднані до проксі-серверу чи надбудованих мереж варто очікувати спад швидкості з'єднання. Апаратні системи захисту, особливо якщо це корпоративна мережа, потребують встановлення окремих серверів, маршрутизаторів, комутаторів та інших додаткових апаратних рішень які, окрім загальної вартості, потребують регулярної перевірки та постійного моніторингу. Найкращим рішенням буде поєднання концепцій загальної мережевої гігієни та програмно-апаратних методів захисту.

1.3. Висновки до розділу 1

Разом з позитивним розвиненням комунікацій людства в те що сьогодні називають «всесвітньою павутиною», розвивалися також і небезпеки користуванням мережею та проблеми захисту. Правоохоронні органи цивілізованих країн докладають багато зусиль щоб користування мережею було якомога безпечнішим а його сегменти дотримувалися законів та правових норм. На багатьох web-ресурсах залишається невидимий слід, іноді

це відбувається за згодою користувача, але і тоді невідомі усі нюанси зберігання та захисту даних.

Основними небезпеками є саме порушення конфіденційності та витік персональних даних. Усі інші є лише похідними від цих двох схожих понять. Так втрата коштів з банку є різновидом витоку персональних даних, тобто персонального логіну та паролю, «деанонімізація» прихованого користувача форуму, навіть якщо в базі даних відсутні персональні дані, є порушенням конфіденційності особи. Перехоплення мережевого трафіку може бути як порушенням анонімності так і витоком даних, оскільки ми, по-перше, компрометуємо персональні дані які передавались цим з'єднанням. По-друге, сам факт передачі таких даних може бути анонімним і його розкриття призводить до порушення анонімності, що певна інформація передавалась. Окремим прикладом є соціальні мережі, де виставлення персональних даних на показ є другою, після спілкування, функцією таких сервісів. Але і в цьому випадку можуть виникнути нюанси – занадто багато особистих персональних даних можуть завдати школи користувачеві, а злам акаунту взагалі призвести до порушення як анонімності так і виставлення на показ усього того що було приховане за персональною сторінкою, тобто такі речі як особисті повідомлення з компрометуючим, небежаним змістом.

Безпека інформації користувачів всесвітньої павутини на сьогодні є чи не основним питанням що часто піднімається як серед повсякдених користувачів, так і серед науковців та державних діячів. На фоні витоків, здавалося б з безпечних, соціальних мереж та поштових сервісів, стандартного пропонованого захисту не вистачає для збереження інформації. Концепція мережевої гігієни може бути справді ефективна лише з поєднанням додаткових засобів захисту.

Функціональність програмних інструментів довзоляє на високому рівні уберегти себе, а їх різноманітність дозволяє обирати, від персонального захисту до захисту цілих корпоративних внутрішніх мереж. Не дивлячись на існуючі загрози, ефективність безпечної роботи в мережі, навіть з

використанням різних додатків та програм, залежить саме від людини. Людський фактор та відсутність базових знать для убереження своїх даних і призводить до більшості негативних наслідків користування, здавалось би, безпечним інтернетом. На жаль багато хто нехтує навіть простими правилами поводження в інтернеті.

РОЗДІЛ 2
ІНСТРУМЕНТАЛЬНІ ЗАСОБИ МЕРЕЖЕВОЇ БЕЗПЕКИ

2.1. Загальна характеристика та класифікація засобів захисту

Через те, що проблеми з безпекою, як і джерела загроз, бувають різними, виникла необхідність в створенні різних видів її забезпечення. Засобами захисту є рішення забезпечення конфіденційності, анонімності та запобіганню витоку даних користувача, та відрізняються різноманітністю застосовуваних методів.

Кожен з видів захисту класифікувати за методом дії та призначенням захисту. Більш широко можна класифікувати на наступні групи:

- Програмні засоби;
- Апаратні засоби;
- Засоби змішаного виду;
- Засоби адміністративного характеру.

Класифікація є більш загальною, так як кожен окремий пункт може мати і інші підвиди захисту, як схожі так і абсолютно різні за видом та призначеннями.

Апаратні засоби дозволяють перешкодити прямому проникненню, допомогти з шифруванням та контролем підключення, та застосовуються для вирішення наступних завдань:

- Моніторинг стану мереж;
- Локалізація збоїв та витоку інформації;
- Протидія несанкціонованому доступу;

- Створення захищеної внутрішньої (локальної) мережі;
- З'єднання вузлів комп'ютерної мережі в межах одного сегменту;
- Шифрування та передача трафіку.

Технічні засоби надійні, незалежні від суб'єктивних факторів і мають високу стійкість до модифікації. Недоліками є ціна на такі пристрої. Встановлення додаткового серверу чи комутатору, особливо для великих мереж що мають сотні пристроїв, може бути дуже затратним. Також вони недостатньо гнучкі і практично завжди мають велику масу та обсяг та потребують додаткового фізичного захисту. Не дивно що серверні кімнати та дата-центри великих підприємств охороняються на рівні захисту фінансових установ.

Програмні засоби оперують різноманітним програмним забезпеченням шифрування, зміни мережі, контролю доступу тощо, і є найбільш поширеними. Такі засоби можуть шифрувати дані та видаляти залишкову інформацію (на зразок тимчасових файлів). Якщо система використовує програмні засоби для захисту, вона отримує масу переваг. Вони гнучкі, універсальні та досить прості в налагоджені, а ще здатні до модифікації та подальше покращення. Їх можна класифікувати наступним чином:

- Вбудовані засоби захисту інформації;
- Антивірусні програми;
- Спеціалізовані пз захисту від несанкціонованого доступу;
- Міжмережеві екрани (брандмауери);
- Проксі-сервери;
- Віртуальні приватні мережі.

Однак цей вид засобів дуже чутливий до випадкових і навмисним змін. До недоліків програмного захисту можна віднести використання частини ресурсів комп'ютера чи сервера, обмежену функціональність мережі, в тому числі спад якості з'єднання, особливо при передачі великих за розміром даних. Такі засоби можуть бути як безкоштовні так і платні, особливо якщо стосується підключення до захисту великих мереж.

Змішаний вид захисту є поєднанням програмно-апаратних засобів, та не обмежується лише одним поєднанням, наприклад, окремого серверу та VPN. В змішаному

Засоби адміністративного захисту включають засоби організаційно-технічного та організаційно-правового характеру. Мається на увазі як онлайн моніторинг так і позамережевий фізичний захист. Сюди можна віднести контроль доступу до приміщень де зберігається мережеве обладнання, розробка політики безпеки та інші засоби не пов'язані з роботою у самій мережі.

Для звичайного користувача підходять саме програмні рішення захисту, встановлення додаткового апаратного пристрою не потрібно для організації комфортного та безпечного домашнього користування інтернетом. Для користування публічними мережами WI-FI чи інтернет-кафе особливо рекомендують користуватися захисним пз.

2.2. Аналіз ефективності існучих засобів захисту безпеки користувачів

Існуючі програмні засоби по різному ефективні для користувача. Одні можна використовувати прямо в браузері, для інших необхідне додаткове пз. Інтернет-користувачі часто довіряють системам що не знаходяться безпосередньо під їх контролем. З появою хмарних обчислень та постійно зростаючим числом сервісів, перенесених в хмару, така ситуація виникає частіше. Такими системами є анонімні та конфіденційні мережі, проксі-сервери та спеціальні анонімайзери.

Базовою системою захисту є брандмауер, або мережевий екран (фаєрвол). Зайзвичай це програмний елемент мережі що виконує контроль та фільтрацію прохідного трафіку. Серед завдань, які вирішують міжмережеві екрани, основними є захист сегментів мережі або окремих хостів від несанкціонованого доступу. Додатково на рівні з брандмауером може працювати класичний антивірус. Сучасні антивірусні засоби мають утиліти призначені для захисту мережі. Класичних засобів захисту може бути

недостатньо через їх обмежену функціональність та ресурсоємкість. Часто антивірусні програми чи брандмауери неправильно інтерперитують підключення чи вплив інших програм, тому можуть ненароком обмежити функціональність прикладних програм та мереж, якими користується користувач, прийнявштх їх за шкідливе пз чи мережеве вторгнення і взагалі не надати доступ. Нерідко, незважаючи за безпеку, в домашньому користуванні такі засоби просто відключають.

Проксі-сервер дозволяє захищати клієнтський комп'ютер від деяких мережевих атак і допомагає зберігати анонімність клієнта. Основна причина використання – забезпечення анонімності та обхід цензури, заборони на відвідування певного ресурсу. Заборона може бути як зі сторони мережі, чи провайдера так і зі сторони ресурсу. На раніх етапах існування всесвітньої мережв, проксі-сервери були найпопулярнішим способом виходу в інтернет з локальних мереж, та і зараз використовується у великих корпоративних мережах різних компаній з цією ж метою. Проксі-сервер є посередником між запитами його клієнта до кінцевої точки (Рис.2.1). Спочатку клієнт з'єднується з проксі-сервером та відбувається запит на певний ресурс, розташований на іншому сервері. Потім проксі-сервер або підключається до вказаного ресурсу, або повертає його з власного кешу (у випадках, якщо проксі має свій кеш). Виділяють окрему категорію проксі-серверів - web-проксі-сервери, що представляють собою web-додатки, встановлені на web-сервері (наприклад на базі Perl, Python чи PHP-скриптів) [3].

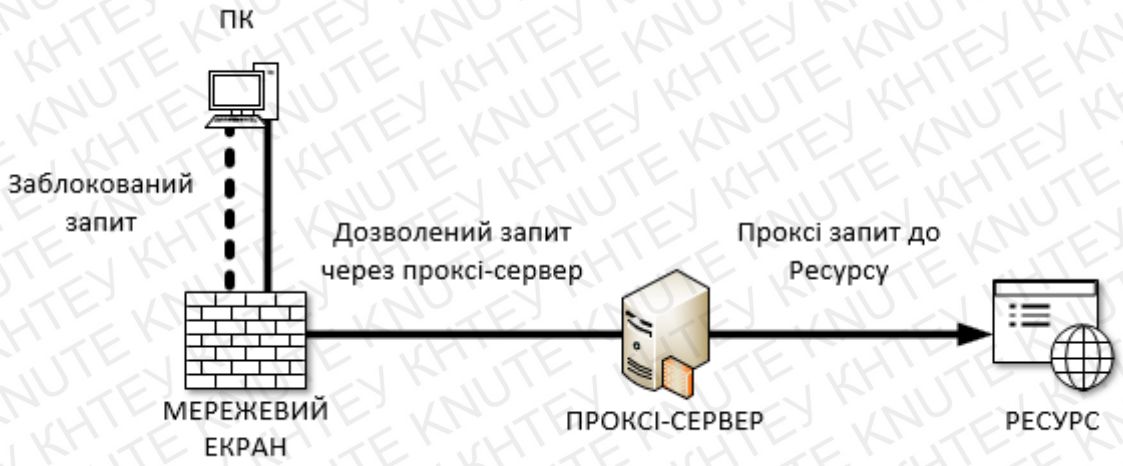


Рис 2.1. Схема доступу до заблокованого ресурсу через проксі-севрер

Ефективне використання проксі можливе і для моніторингу та фільтрації контенту. Сервер що фільтрує вміст, забезпечує адміністративний контроль над вмістом, що може бути переданий у різних напрямках через проксті. Таке використання має місце як в комерційних, так і некомерційних організаціях для забезпечення відповідності політиці використання. Проксі-сервери фільтрації часто підтримують автентифікацію користувача для контролю доступу до інтернету. Певний ряд факторів що може впливати на ефективність роботи з проксі-серверами. Так доступ через проксі на деяких ресурсах може бути заборонений, IP що відправляється сервером може бути занесений в список так званих небезпечних IP-адресів.

VPN, або віртуальна приватна марежа є схожим за функціоналом з проксі-сервером методом. В той час як проксі-сервер використовує саме сервер, віртуальна приватна мережа є узагальненням мереж що створюються поверх інших мереж, які мають менший рівень довіри. Концепція віртуальної приватної мережі, більш відома як VPN, з'явилася як фінансова альтернатива захищеної комунікації через громадські канали зв'язку, такі як інтернет, і незабаром стала технологією, послугою, орієнтованою на безпеку, що гарантує цілісність, конфіденційність і автентичність відбитку інформації.

VNP тунель створюється між двома вузлами, дозволяє клієнту бути повноцінним учасником віддаленої мережі, користуватися внутрішніми сайтами, базами та політиками безпеки. Зазвичай VPN утворюють на рівнях не вище мережевого, так як застосування криптографії на цих рівнях дозволяє використовувати в незмінному вигляді транспортні протоколи такі як TCP чи UDP. Підключення здійснюється за допомогою розширень браузеру, web-сервісами або програмами. Різні технології VPN використовуються з різним ступенем шифрування. Наприклад, протокол тунелювання «точка-точка» (PPTP) працює швидко, але набагато менш безпечний, ніж інші протоколи, такі як IPSec або OpenVPN, який використовує SSL / TLS (Secure Sockets Layer / Transport Layer Security). Ще один спосіб поліпшити безпеку VPN - це досконала пряма секретність (PFS). Якщо використовується PFS, зашифровані повідомлення та сеанси, записані раніше, можуть бути отримані та дешифрувати якщо скомпрометовані довгострокові секретні ключі або паролі.

З PFS кожен сеанс VPN використовує різну комбінацію ключів шифрування, тому навіть якщо буде скомпрометовано один ключ, інші сеанси VPN неможливо буде розшифрувати [5].

І VPN, і проксі-сервер отримує доступ до будь-якого ресурсу в інтернеті, не розкриваючи при цьому свій реальну IP-адресу та місцезнаходження. Однак послуги дещо відрізняються, і це слід враховувати. При підключенні до VPN, усі користувачі можуть використовують єдину IP адресу. Одночасне використання IP-адрес використання підключення через VPN, роблячи його придатним для інтернет-серфінгу і збереження анонімності, але марним для роботи з акаунтами і деякими програмами, оскільки всі акаунти, що працюють з одного VPN часто блокуються політиками безпеки.

В обох випадках захищена передача може впливати на швидкість з'єднання. Наприклад, швидкість з'єднання залежить від віддаленості проксі сервера, чи обраного регіону VPN з’єднань. Вибір технології та і методів

шифрування повинен проводитися в кожному конкретному випадку, залежно від того, які дані будуть передаватися. Наприклад проксі вирішують більше завдань, коштують дешевше, працюють швидше і стабільніше. В той же час підключення до VPN швидше, та дозволяє швидко змінювати мережу (регіон). В обох випадках передача даних може якісно шифруватися.

Найбільш складними за архітектурою є поєднання подібних мереж у повноцінну «цибулеву маршрутизацію» чи надбудови зашифрованих анонімних мереж над вже існуючими. Найвідоміші приклади таких мереж – I2P, GNUnet, Freenet та The Onion Router (Tor), використовують саме надбудови над існуючими мережами (інтернет) до яких неможливий доступ ззовні без використання спеціального програмного забезпечення. Такі анонімні мережі називають «даркнет» та вважаються найефективнішим інструментом забезпечення виского рівня приватності користувачів в інтернеті.

Найбільш популярною та широковідомою є система Tor. Tor володіє трьома досить важливими властивостями, що забезпечують таку високу захищеність мережі. Система працює за рахунок децентралізації вузлів, тобто відсутність централізованого серверу чи клієнта (Рис. 2.2.). Другою властивістю Tor є оверлейність – робота поверх будь-якої іншої мережі, як інтернет так і іншої анонімної мережі. Третьою властивістю є так звані волонтерські вузли – будь-яка людина на планеті може надати вузол для Tor, сприяючи розростанню мережі і збільшення її пропускної здатності. Таким чинон, чим більше вузлів знаходиться всередині мережі, тим сильніше захищена анонімність користувача [1].

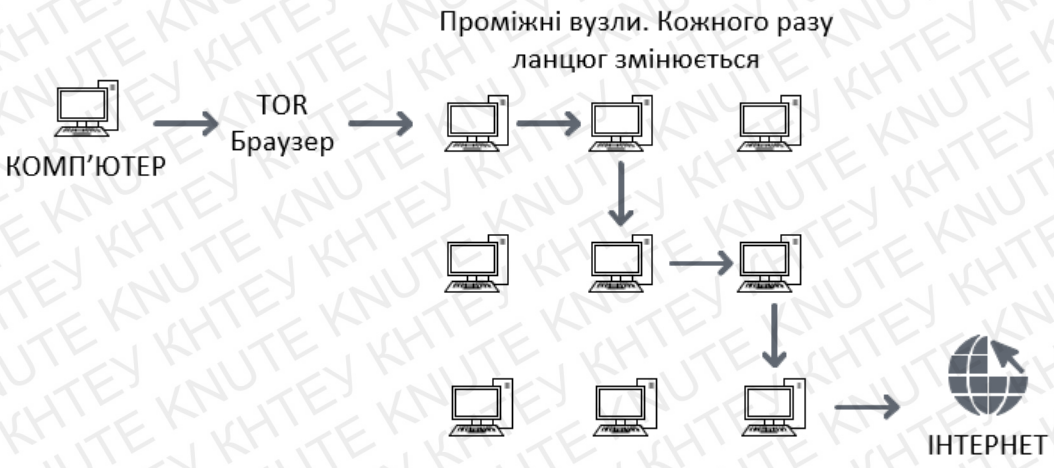


Рис. 2.2. Загальний принцип роботи TOR

Такі системи не рідко використовуються в обхід цензури чи інших блокувань, навіть державного рівня. Багато людей не згодні миритися з контролем з боку держави та блокуванням певної інформації, в тому числі ідеологічної. Наприклад Китай забороняє VPN, Tor та інші засоби шифрування і вимагають розкривати свою особистість для будь-якої активності в інтернеті. Зі схожими методами Російська федеральна служба з нагляду в сфері зв'язку (Роскомнадзор), підтримла ідею заборони анонімайзерів, та намагається призупинити поширення подібної по суті інформації про способи обходу заблокованих сайтів [2]. Використання таких систем більш для грубого порушення закону явно є негативним аспектом, з яким боряться навіть розробники таких систем, особливо якщо злочини не стосуються звичайного обходу блокування.

2.3. Розробка моделі захищенного інтернет-серфінгу

Під інтернет-серфінгом розуміється перегляд web-сторінок шляхом переходу з однієї web-сторінки на іншу за допомогою гіперпосилань. Захищена модель передбачає користування інтернетом з мінімальними ризиками компрометації даних користувача. Завданням моделі є захист користувачів від загроз, під час інтернет-серфінгу в повсякденному користуванні, як в домашньому так і робочому середовищі (якщо це є

частиною робочого процесу і не відволікає від роботи). Особливо така модель необхідна особам що з певних причин мають приховувати, частково або повністю, свої персональні дані, наприклад публічні особи чи державні діячі.

Модель містить реомендації використання пз додаткового захисту, з комбінацією правил мережевої гігієни. Узагальнити рекомндації можна наступним чином:

- Дотримуватися правил мережевої гігієни;
- Використовувати додатковий шар захисту між браузером та мережею;
- Вчасно оновлювати пк;
- Використовувати перевірени ресурси пошуку чи збереження інформації.

Дотримання правил мережевої гігієни важливо і при використовувані додактового програмного чи web-сервісного захисту. Ігнорування базових правил безпеки все ще загрожує користувачеві програмно-апаратних засобів.

Інтернет-серфінг часто передбачає швидкий перехід між посиланнями, і, згадуючи підставні фішингові web-сайти, потрібно дотримуватися обережності, так як ні VPN мережі, ні Proxy-сервіси не захищають від шкідливого пз що можна підхопити зайвий раз завантажуючи файл перейшовши за нібето безпечним посиланням.. Сучасні web-браузери блокують шкідливі адреси але їх база даних оновлюється вже після виявлення існуючих загроз, до того ж на певного користувача особисто може спеціально бути направлена фішингове розсилання.

Між браузером та мережею має використовуватися один із способів захисту анонімності та персональних даних. Це може бути встановлення окремого програмного забезпечення, розширення браузера чи використання особливих web-ресурсів (Рис. 2.3). Так сучасні браузерами можуть мати

вбудований механізм захисту, наприклад безкоштовний сервіс VPN від Opera, чи додаткові розширення Google Chrome (FreeVPN, HolaVPN). Користування спеціальзованими програмами більш надійне, та може шифрувати трафік ще до використання браузера, але такий підхід не є зручним для швидкого інтернет-серфінгу.

Рис. 2.3. Схема безпечного web-серфінгу через захищений ресурс

Так захист відбувається прямо в браузері не навантажуючи ресурси ПК та не застосовуючи сторонні програми. Вибір web-ресурсу чи програмного забезпечення залежить від уподобань користувача та кількості трафіку що має кріз нього проходити. Такий же підхід в захисті навпаки може використовувати сам web-ресурс, тобто користувача під'єднують спершу до проксі сервера, звідки йде «дзеркало» на запитуваний ресурс.

Альтернативний варіант – використання саме спеціальних браузерів, на кшталт TOR Browser, але такий захист може блокувати і необхідний користувачеві контент, наприклад елементи JavaScript, та бути занадто повільними через децентралізацію вузлів, що може лише завдати дискомфорт, наприклад, при перегляду відео чи користуючись соціальними мережами.

2.4. Висновки до розділу 2

Інструментальні засоби мережевої безпеки є потужним захистом вільних мереж, навіть якщо користування передбачає звичайний web-серфінг. Якщо про загальний вид загроз як витік персональних даних чи деанонімізація в цілому, вже було наведено, то їх реалізація значно ширша та може включати як програмно-апаратні методи так і соціальну інженерію.

Якщо другий варіант залежить тільки від дій самого користувача, то стати жертвою витоку даних через злам чи сніфінг значно складніше якщо подбати про персональну інтернет безпеку.

Програмні засоби та онлайн сервіси надають різні послуги, але мають спільну мету забезпечення захисту клієнтів. Небажане рекламне відслідковування лише найменша проблема. Сьогодні, здавалося б, звичайна рекламна аналітика може впливати на споживача не тільки заохочуючи до того чи іншого товару. На людях вже відпрацьовані онлайн механізми глобального впливу через такі системи, особливо під час політичних подій. В цифрову епоху на жаль залишаються і такі поняття як переслідування за цифровими даними, постами у соціальних мережех, цензура, тощо.

У розділі були проаналізоваі існуючі засоби захисту користувача, такі як VPN, Proxy та TOR-сервіси. Також було складено загальну модель захищеного інтернет-серфінгу. Захист інформації користувачів стала однією з галузей, для якої розробляються спеціальні інструментальні засоби, створюються вимоги до безпеки IT та оцінюється рівень їх виконання. Ефективність таких методів вже доведена. В деяких країнах доступ до всесвітньої мережі можливий саме засобами VPN та проксі, уникаючи переслідування та цензури. У вільних мережах, під час інтернет-серфінгу, це допомагає захистити користувача та його анонімність, чи вберегти персональне листування. Хоча існує досить розповсюджена думка що системи TOR, I2P тощо, використовуються переважно зловмисниками, саме такі надмережі є ідеальним захистом при транзакціях криптовалюти, та анонімному спілкуванню. До того ж там можна знайти багато наукових робіт та вільних бібліотек.

Проблематика захисту буде підніматися частіше з розвитком новіших технологій. Вже зараз ідуть активні дослідження використання систем блокчейну, у космос запускаються супутники для покриття вільним інтернетом усієї планети а стандарти мобільного зв’язку змінюються на

швидкісний 5G. Розвиток систем захисту повинен виконуватися прямо пропорційно науково-технічному прогресу, а розробка нового програмного забезпечення та методик захисту користувачів всесвітньої мережі є важливим завданням спеціалістів кібербезпеки. На щастя, на фоні світових подій, скандалів, повідомлень про витоки інформації чи навпаки, обмеження доступу, динаміка покращення безпеки рядових користувачів зростає.

РОЗДІЛ 3

РОЗРОБКА ТА ВПРОВАДЖЕННЯ WEB-СЕРВІСУ ЗАХИЩЕНОГО ІНТЕРНЕТ-СЕРФІНГУ

3.1. Вибір програмних засобів та мови програмування

Web-сервісом (або web-службою) перш за все називають технологію зі стандартизованим інтерфейсом що відображається, та здійснює свою програмну функцію саме в браузері, найчастіше у форматі web-сторінки. Таким сервісом називають послуги що надаються в інтернеті, наприклад web-пошта, пошукова система, сховище даних тощо. Як і в розробці звичайного прикладного пз, для побудови web-сервісу необхідно обрати необхідні програмні засоби, чи то платформи та поставаити певні завдання.

Для первинного налагодження серверу буде використовуватися технологія Apache. Apache HTTP Server це технологія відкритого web-серверу, доступна запуску як на Linux-дистрибутивах так і Microsoft Windows. Apache використовується для передачі через HTTP статичних та динамічних web-сторінок у всесвітній павутині. Багато web-застосунків спроектовано, за допомогою можливостей що цей web-сервер. На даний момент є найпопулярнішою в інтернеті технологією web-сервера.

Базові можливості дозволяють оброблювати різноманітні файли конфігурації, протокол HTTP та можливість завантажувати різноманітні модулі що розширюють базові функції. Частина з них розробляється командою розробників Apache Software Foundation, хоча більшість окремими open-source розробниками. Модулі можуть бути включені в сервер в момент

компіляції, або завантажені динамічно, через конфігураційний файл. В модулях реалізуються такі речі, як підтримка мов програмування, додавання функцій, виправлення помилок або модифікація основних функцій, посилення безпеки тощо [8].

Мовою програмування та налагодження обрано PHP: Hypertext Preprocessor, або просто PHP. Це скриптова мова програмування для роботи та генерації HTML-сторінок на стороні web-серверу, є однією з найпроширеніших мов які використовуються у web-розробках. При використанні сервер інтерпретує код PHP у HTML що передається на сторону клієнта. Користувач не бачить самого PHP на сторінці, тому що браузер отримає лише HTML код, в той же час завдяки PHP виконуються задані скрипти чи перетворення з якими уже і взаємодіє клієнт. Усю документацію та пояснення можна знайти як на офіційному сайті, так і на сотнях інших інтернет-ресурсів [9].

Серед основних властивостей PHP можна виділити:

- Висока швидкість роботи;
- Багатоплатформеність;
- Різноманітні інтерфейси та бібліотеки для роботи з базами даних;
- Вільне поширення;
- Простота застосування при широких можливостях;
- Висока ефективність у створенні web-застосунків.

PHP належить до інтерпретованих мов програмування, тобто код не перетворюэться попередньо у повнісью машинний, а виконується рядок зя рядком за допомогою інтерпретатора. Це дозволяє обробляти сценарії з достатньо високою швидкістю.

Різноманітність PHP дозволяє уникати написання багаторядкових функцій, призначених для користувача як це відбувається, наприклад, в C або Pascal. Код PHP дуже схожий на той, який зустрічається в типових програмах мовами C, Perl або Pascal, та має запозичені з них конструкції мови. PHP

виконує код, що знаходиться в середині обмежувачів <?php код ?>. Все, що знаходиться поза обмежувачами, виводиться без змін. Таким чином виконується вставка PHP-коду в HTML-код.

За основу буде взято рушій PHP-Proxy. Це вільна інтерпретація PHP коду під потреби проксі серверів корпоративних мереж та сервісів. В цій розробці він буде застосовуватися для підтримки проксі функцій з виводом захищених сторінок користувачеві [10]. За необхідності до нього можна написати та підключити певні плагіни. Для підключення завантаження та менеджменту рушія використаємо Composer. Це менеджер пакетів мови програмування PHP, він забезпечує можливість завантаження файлів та біблікотек, і встановлення залежностей між файлами PHP.

Для написання та редагування коду буде використовуватися текстовий редактор Sublime Text. Він підтримує велику кількість мов програмування, має зручний інтерфейс. Незалежно від мови програмування, коли користувач набирає код редактор автоматично пропонує варіанти для завершення запису, підсвічує синтаксис та дозволяє будувати і запускати програми без необхідності в командному рядку.

Збереження проекту та доступу до нього через мережу інтернет буде використовуватися хостинг 000webhost наданий компанією Hostinger. Далі буде зареєстроване власне доменне ім’я для доступу до серверу. Для швидкої роботи з віддаленими файлами використаємо FTP-клієнт FileZilla.

3.2. Загальна характеристика та спосіб функціонування системи

web-сервіс має на меті стати посередником між браузером та інтернет з’єднанням. Це web-проксі побудовнаий на PHP-Proxy рушієм-бібліотекою з виводом у вгляді web-сторінки. На титульній сторінці буде доступне вікно введення адреси та додатковий вибір дзеркал перенаправлень, тобто вибір інших серверів с аналогічною функцію, але з іншими IP адресами та фізичним розташуванням. При використанні сервісу користувачеві буде

запропоновано ввести адресу необхідного сайту. Після обробки на екрані з’явиться ресурс, адресу якого було введено, але з додатковим вікном сервісу та з’єднанням через обраний проксі сервер. Так користувач отримує доступ до запитуваного ресурсу, але саме з'єднання буде шифроване додатковим шаром, а користувач не буде розкривати свій реальний IP чи розташування. Подальший захист залежить від принципів мережевої гігієни. На малюнку представлена схема реалізації web-серверу як засобу захисту з’єднання користувача під час інтернет-серфінгу (Рис. 3.1).

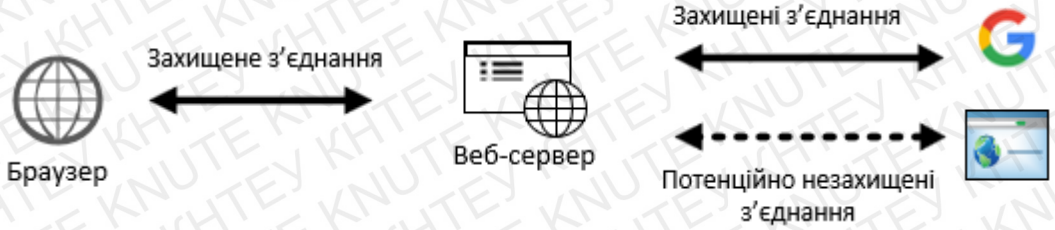


Рис. 3.1. Захист з'єднання через проміжний вузол.

3.3. Реалізація та впровадження системи

Для початку налагодимо локальний сервер Apache з підключенням PHP для первинного тестування та налагодження. Створимо структуру каталогів серверу (Рис. 3.2). Так розділимо файли серверу та файли сайтів з базами даних. Це зручно для обслуговування серверу, тестування та резервного копіювання.

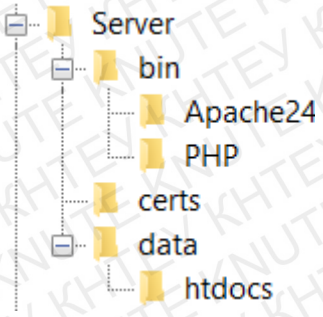


Рис.3.2. Первинна структура каталогів

Інсталюємо середовище Apache. Для цього вивантажемо офіційний дистрибутив з web-сайту https://apachelounge.com/download за C:\Server\bin\.

Після завантаження привя’жемо сервер до нашого шляху. Відредагуємо файл httpd.conf з посиланнями на c:/Server/bin/Apache24. Відредагуємо конфігурацію Define SRVROOT з параметрів "c:/Apache24" на "c:/Server/bin/Apache24", а <Directory "${SRVROOT}/htdocs"> та DocumentRoot на "c:/Server/data/htdocs/" та додамо index.php в змінну DirectoryIndex. Так ми прив’яжемо Apache саме за нашою адресою. За замовчуванням можна використовувати і шлях C:/Apache24, але для окремого проекту та підключення додаткових модулів на кшталт PHP чи MySQL, краще використовувати окрему директорію.

Запустимо Apache сервер для перевірки роботи через командний рядок в режимі адміністратора, написавши c:\Server\bin\Apache24\bin\httpd.exe -k install, та c:\Server\bin\Apache24\bin\httpd.exe -k start. Якщо налаштування правильні отримаємо наступну сторінку при введені localhost в адресну строку браузеру (Рис.3.3.).

Рис. 3.3. Перевірка роботи серверу Apache

Далі встановимо та налаштуємо PHP. В директорію c:\Server\bin\PHP завантажимо PHP файли з офіційного сайту розробника. В попередньому файлі httpd.conf додамо наступні налаштування прив'язки PHP та завантаження PHP модулю Apache - PHPIniDir "C:/Server/bin/PHP", AddHandler application/x-httpd-php .php та LoadModule php7_module "C:/Server/bin/PHP/php7apache2_4.dll".

З офіційної сторінки завантажимо PHP-proxy бібліотеку. Це можна зробити як вручну так і за допомогою менеджеру Composer. Інсталюємо її в

C:\Server\data\htdocs, створивши папку engine. Бібліотека також підтримує написання різноманітних плагінів для роботи з різними мережами та сторінками. Бібліотека складається з наступних файлів:

- Файл Config.php – основна конфігурація рушія;
- Файл helpers.php – конфігурація роботи с текстовими даними та шифруванням;
- Файл Html.php – конфігурація роботи з html форматом;
- Файл Proxy.php – робота з самими з'єднаннями;
- Директорія athlon1600\php-proxy\src\Http з місткістю конфігурацій роботи з HTTP;
- Директорія athlon1600\php-proxy\src\Plugins з набором стандартних плагінів;

Сервіс має працювати у вигляді web-сторінки, тому бібліотека має бути підлаштована під окремий сайт. Створимо файл index.php. Це скрипт під'єднання до рушія та виводу форми вводу web-адреси для користувача. Скрипт буде перевіряти наявність файлів рушія та звертатися до них за для передачі даних. Для рушія теж проведемо необхідні зміни, для роботи з web-сторінками. Створимо файл config.php для додаткової конфігурації попереднього файлу та можливості налаштування ключа шифрування. За допомогою цього конфігураційного файлу під'єднаємо стандартні плагіни доступу cookie файлів та url-форми. Створимо окрему директорію forms куди помістимо саме сторінки вводу та виводу інформації. Файл main.php буде відповідати за стартову сторінку яку бачить користувач, та де вводить необхідний йому web-ресурс. Сторінка має поле для введеня посилань, текстову інформацію про автора та використовуване пз, а також можливість зміни зеркала з'єднання. Файл url_form.php призначений для роботи та відображення посилань коли таке з'єднання вже виконано. Він додає форму для зміни посилання прямо на сторінці уже під'єднаного ресурсу угорі, а також кнопку повернення на головну форму.

В даному випадку поєднуються технології PHP та мови HTML. Отримані з форм дані передаються на index.php, звідки транслють запит на рушій та в інтернет-мережу. Лістинг коду цих сторінок з поясненями надано в Додатку В.

Загальну структуру роботи можна відобразити на схемі (Рис 3.4.). Після доступу до ресурсу завантажується index.php що попередньо регулюється config.php, та піключається до бібліотеки PHP-Proxy, після чого транслюється сторінка вводу посилань, main.php яка і забезпечує клієнту відображення необхідних даних.

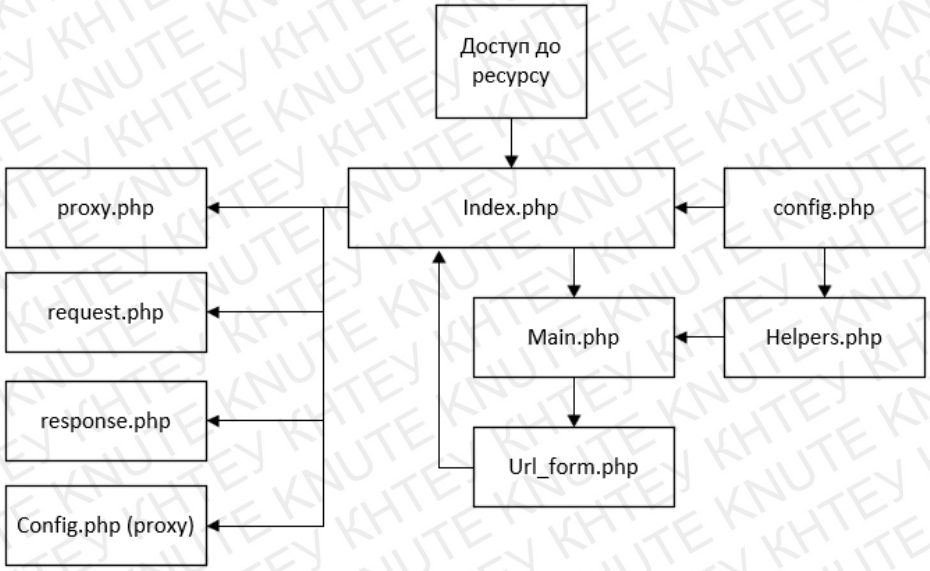


Рис.3.4. Схематична робота web-сервісу

Перевіримо роботу на локальному Apache сервері. Відкриємо браузер та введемо localhost, після чого отримаємо оформлену стартову сторінку (Рис. 3.5).

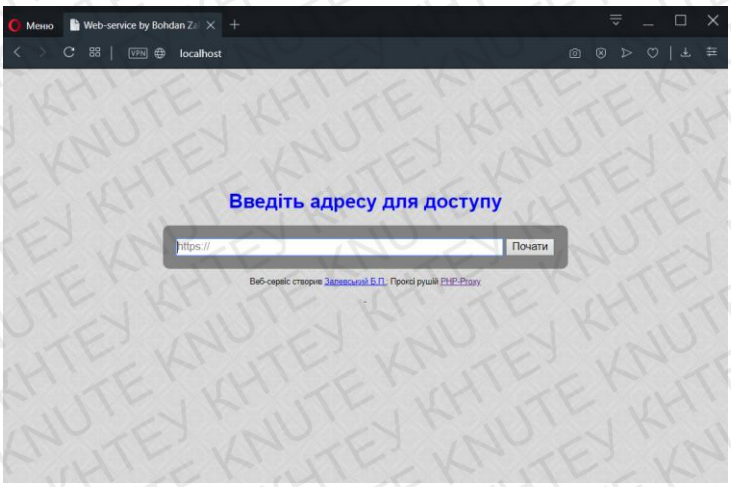


Рис.3.5. Стартова сторінка web-сервісу

Перевіримо функціональність. Для переходу www.google.com (Рис.3.6).

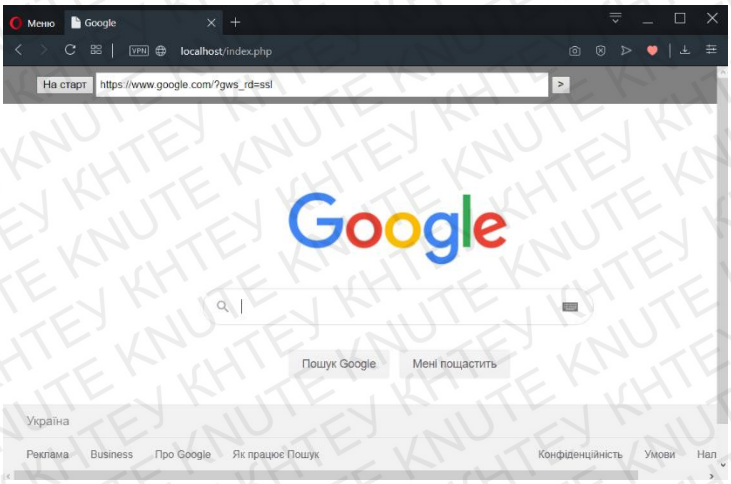


Рис. 3.6. Функціональність web-сервісу на прикладі доступу до google.com

Тепер коли локально перевірена функціональність розробки необхідно викласти її на інтернет хостинг та прив'язати доменне ім'я. Справа в тому що локальний сервер є локальним, тому проксі з'єднання йде на той же сервер звідки його запущено, і для повного функціонування варти мати онлайн сервер. В якості хостингу буде виступати 000webhost з можливістю безкоштовного зберігання та преєднання доменних імен. Для доступу варто лише зайти на сторінку www.000webhost.com, зареєструватися і почати працювати. Після створення сайту під'єднуємо його до файлового менеджеру FileZilla, для завантаження файлів web-сервісу (Рис. 3.7).

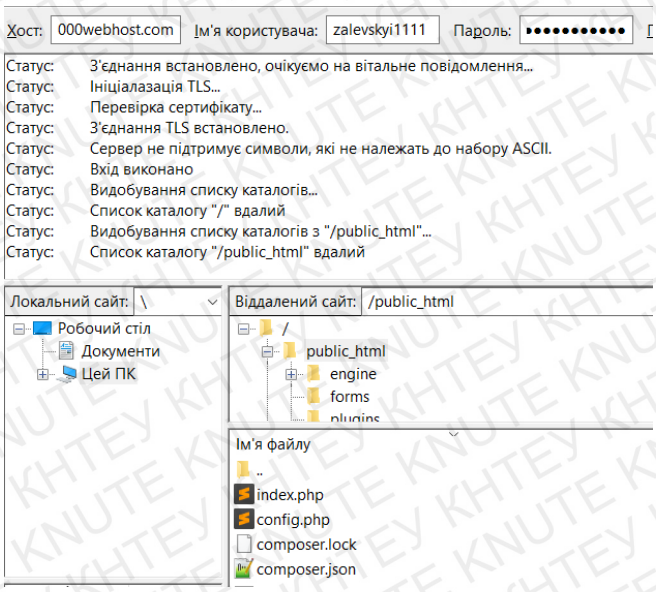


Рис.3.7. Файли вивантажені на віддалений хост

Для зручності використання було придбано домен protectsurf.xyz що перенаправляє на один із серверів. Окремі адреси можна прописати в ручну:

- https://zalevskyi1111.000webhostapp.com
- https://zalevskyi2222.000webhostapp.com
- https://bz1111.000webhostapp.com
- https://bz2222.000webhostapp.com

Різні сервери дають різні IP адреси, відповідно використання web-сервісу приховує особу та захищає від небажаного відслідковування. Для зручності, на головну сторінку було додано елемент випадаючого списку для швидкої зміни серверу (Рис.3.8).

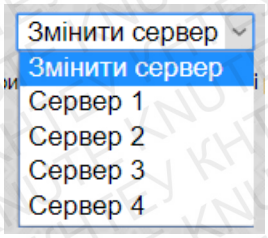


Рис.3.8. Список серверів

Перевіримо сервіс викладений на хостинг (Рис.3.9).

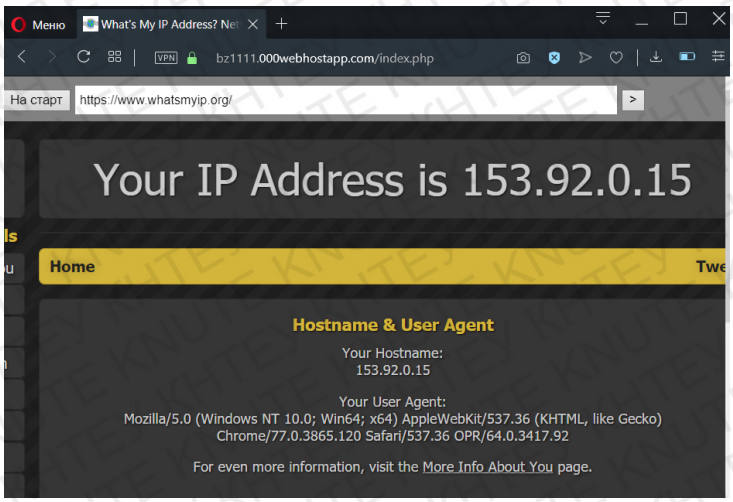


Рис.3.9. Змінений IP під час користування.

3.4. Висновки до розділу 3.

Проаналізувавши сучасне програмне забезпечення, було прийнято рішення використовувати Apache для локального налагодження сервісу, як потужний і в одночас зрозумілий сервер. В якості мови програмування було вибрано PHP завдяки його скриптовій структурі, швидкості виконання команд та легкою інтерграцією у web-контент. Розроблений додаток може працювати як автономно на виділеному сервері за допомогою Apache, так і в мережі інтернет, використовуючи віртуальний хостинг.

Web-сервіс допомагає сховати IP адресу кінцевого користувача, перенаправлючи через сервер необхідний інтернет контент. В процесі розробки була використана PHP-Proxy бібліотека захисту домашніх та приватних мереж, але у вигляді самого web-сервісу, ігноруючи встановлення додаткового пз.

Сервіс має під собою функціонал прийняття заданої інформації у вигляді посилань, обробкою через проксі з'єднання та візуалізацією кінцевому користувачу у вигляді захищеної web-сторінки, з можливістю зміни чи збереження інформації. Захист відбувається прямо в браузері

користувача, без необхідності додаткового налаштування сторонніх програм. Для доступу в онлайн режимі був використаний хостинг 000webhost та реєстратор доменів reg.ru, де було зареєстровано домен protectsurf.xyz. Завдяки цьому було реалізовано декілька серверів з різними IP адресами, та функціоналом перемикання між ними.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В результаті виконання кваліфікаційного проекту, було проведено аналіз сучасних загроз даних користувачів, з виділенням в окремі категорії порушення конфіденційності і витік даних. Також було проаналізовані сучасні засоби захисту як у вигляді класичних програмних засобів, так і окремих мереж чи web-ресурсів.

Згідно тематики кваліфікаційної роботи, був розроблений та впроваджений web-сервіс захищеного інтернет-серфінгу, використовуючи технології Apache та мову програмування PHP. З допомогою розробленого сервісу можна безпечно користуватися мережею інтернет з захистом конфіденційності та від відслідковування. Сервіс, та сервери що його містять, пройшли тестування в використанні на різних інтернет-браузерах, при завантаженні різних за навантаженням сторінок.

Паралельно була змодельована сама модель захищеного інтернет-серфінгу через додатковий, програмний або мережевий шар захисту, та загальні рекомендації мережевої гігієни. Дану інформацію варто поширювати більш активно, та в подальшому оновлювати відповідно до світових тенденцій захитсу користувачів в мережі.

Використовуючи сучасні мови програмування, на сьогодні розроблено багато потужних систем захисту користувача та з'єднань. В майбутньому питання розробки захисного програмного забезпечення та криптозахисту буде стояти напрочуд гостро, так як з появою нових технологій ростуть як і ризики в помилках таких систем, так і технології умисної шкоди.

Перспективи подальшої розробки мають полягати в динамічних системах захисту, з використанням технологій нейромереж та штучного інтелекту, наприклад динамічне перемикання серверів залежно від навантажень, чи системи автоматичного схову користувача, при загрозі небажаного спостереження. Використання захищених проксі-серверів чи VPN мереж мають стати стандартом при користуванні інтернетом навіть під час домашнього користування. Представники міжнародних організацій та уряду мають більш глибоко вивчати цю проблему, та бути готовими до негативних подій у кіберпросторі, і зосереджувати особливу увагу на допомозі провідним інженерам програмного забезпечення для побудови комплексних систем захисту мільярдів користувачів всесвітньої мережі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Статті/тези, опубліковані у збірниках наукових праць/конференцій.

1. Лазаренко А.В. Технології деанонімізації TOR. / Нові інформаційні технології в автоматизованих системах: матеріали дев'ятнадцятого науково-практичного семінару. ІПМ ім. М.В. Келдиша, 2016. с. 257
2. 7Залевський Б.П., Проблеми анонімності в інтернеті: матеріали Всеукраїнської науково-практичної конференції Безпека соціально-економічних процесів в кіберпросторі / Залевський Б.П., Пашорін В.І. – К.: Київ. КНТЕУ, 2019 – с. 53-54

Журнальні статті.

3. Маркін Д.О., Архипов П.А., Галкин А.С. Дослідження стійкості анонімних мереж на основі web-проксі. / Журнал «Вопросы киберпезопасности». - 2016, с. 21-26

Багатотомні видання.

4. Николахин А.Ю, Використання технології VPN для забезпеченя інформаційної безпеки. / «Экономика и качество систем связи». 2019. Том 2. с. 60-63

Електронні ресурси.

5. Zack Whittaker. Millions of Instagram influencers had their contact data scraped and exposed // TechCrunch, startup and technology news. Дата оновлення: 20.05.2019. URL: https://techcrunch.com/2019/05/20/instagram-influencer-celebrity-accounts-scraped/ (Дата звернення: 20.09.2019).
6. Кіберполіція викрила злочинну групу у продажі персональних даних громадян // Офіційний сайт Департаменту кіберполіції Національної поліції

України. Дата оновлення: 16.10.2018. URL: https://cyberpolice.gov.ua/news/kiberpolicziya-vykryla-zlochynnu-grupu-u-prodazhi-personalnyx-danyx-gromadyan-3440/ (Дата звернення: 20.09.2019).

7. Aircrack documentation // Aircrack-ng tools to assess WiFi network security. URL: https://www.aircrack-ng.org/doku.php?id=Main (Дата звернення: 25.09.2019).
8. Apache Documentation // Apache HTTP Server official page. URL: https://httpd.apache.org/docs/ (Дата звернення: 07.10.2019).
9. PHP Documentation // PHP: Hypertext Preprocessor. URL: https://www.php.net/docs.php (Дата звернення: 07.10.2019).
10. PHP-Proxy library // PHP-Proxy library. GitHub.com URL: https://github.com/Athlon1600/php-proxy.git (Дата звернення: 07.10.2019).

ДОДАТКИ

Додаток А

ПРОГРАМА ТА МЕТОДИКА ТЕСТУВАННЯ

Тестування програмного забезпечення - перевірка відповідності між реальною і очікуваною поведінкою програми, що здійснюється на кінцевому наборі тестів, обраному певним чином. Цілями тестування є перевірка правильної роботи додатку.

Для перевірки правильної роботи сервісу використаємо метод дослідницького тестування. Дослідницьке тестування (Exploratory testing) – це одночасне вивчення програмного продукту та його тестування.

Перевіримо роботу web-сервісу посиланням на пошуковий сервіс. Це буде більш доречно враховуючи можливості пошуку в інтернеті. Для перегляду використаємо інтернет-переглядач Opera (Рис.А.1).

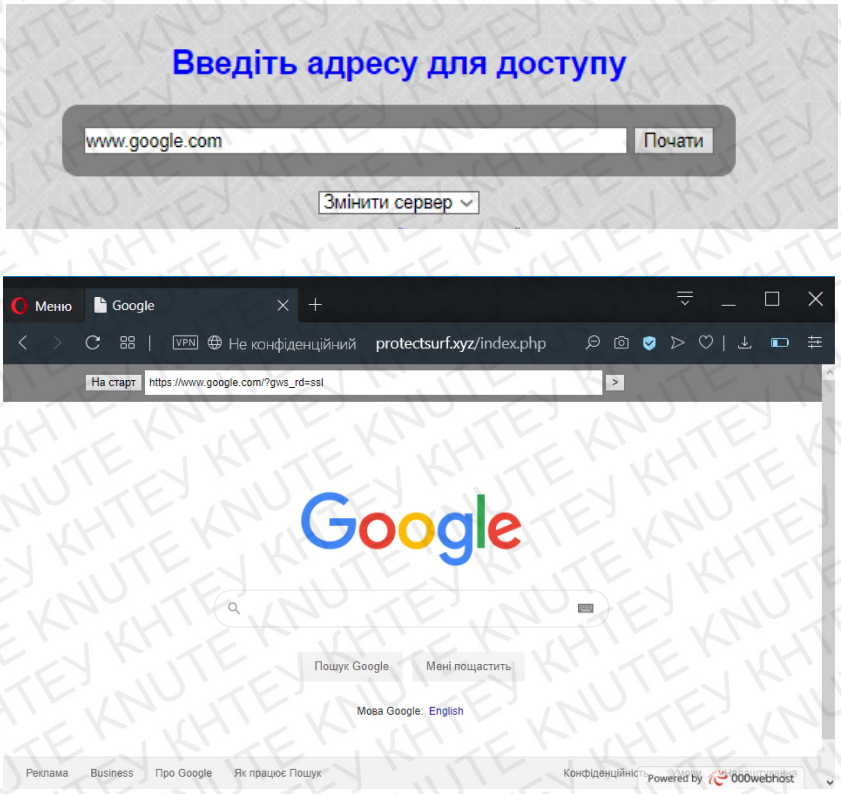


Рис.А.1. Перевірка роботи головної сторінки

Перевіримо функціональність самого сайту. У вікні пошуку введемо випадковий запит (Рис.А.2).

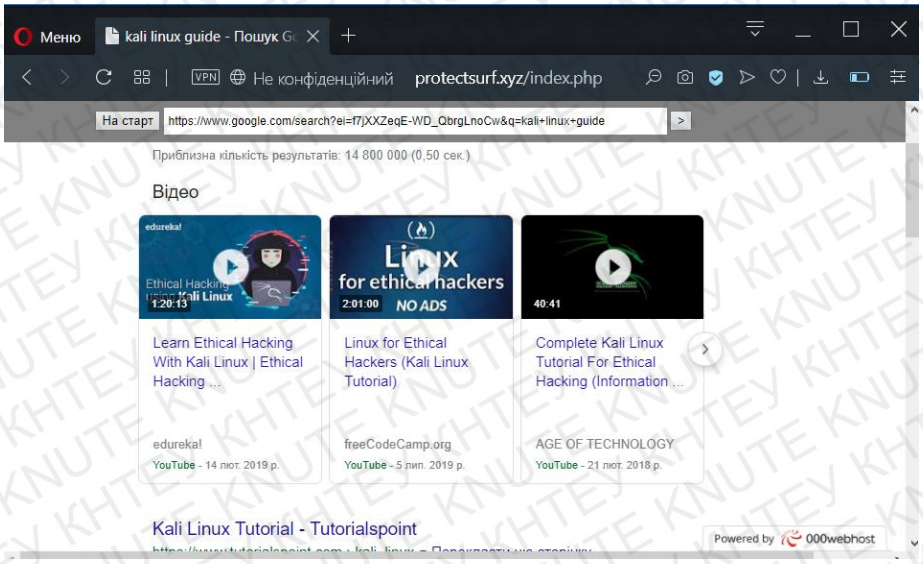


Рис.А.2. Перевірена функціональність внутрішнього сайту

Під час використання сторінка змінила свій вигляд для доступу до контенту, та отримала верхнє меню для швидкої навігації. Перевіримо його функціональність змінивши адресу (Рис.А.3).

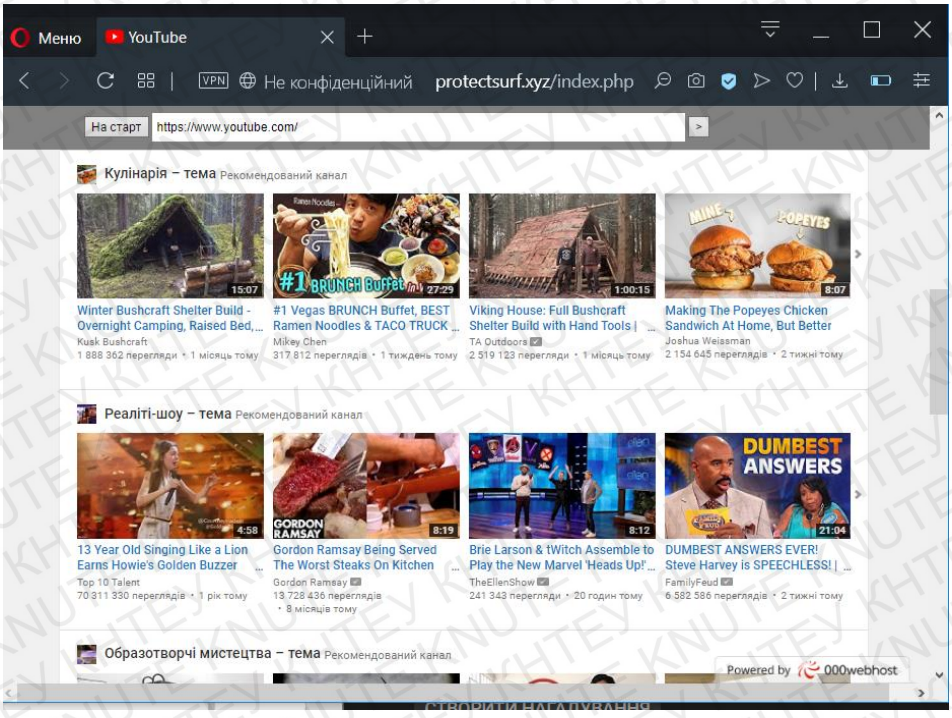


Рис.А.3. Перевірка роботи верхнього меню.

Перевіримо роботу мультимедійних елементів (Рис.А.4).

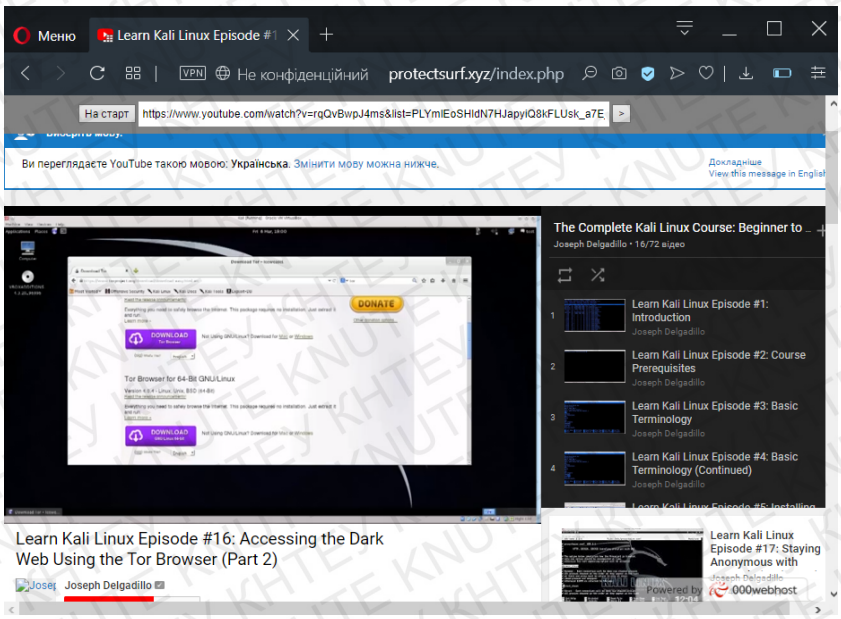
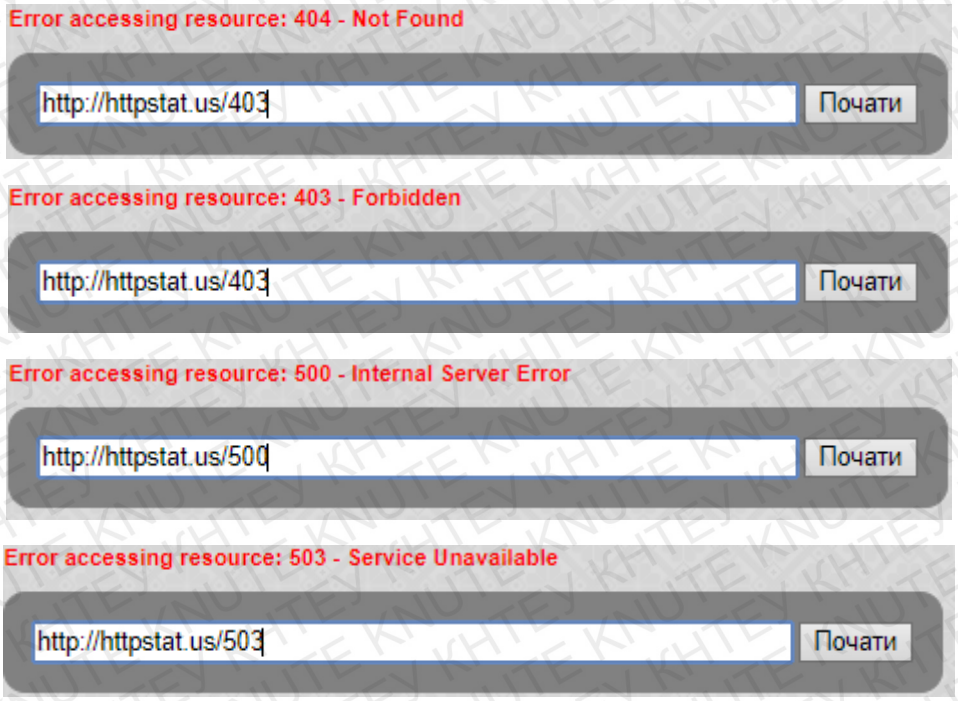


Рис.А.4. Перевірка мультимедійних елементів.

Сервіс має підтримувати відображення HTTP кодів стану у випадку коли помилок. Перевіримо найрозповсюджуваніші з них, а саме 404, 403, 500, 503, 504 за допомогою сервісу httpstat.us (Рис.А.5).



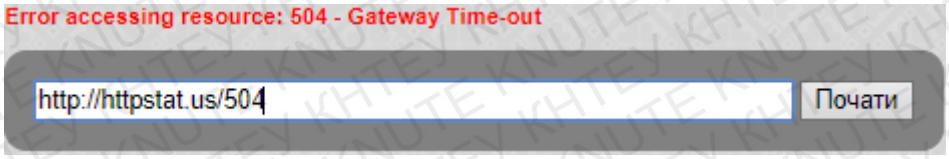


Рис.А.5. Перевірка працездатності кодів стану.

Головне завдання сервісу не тільки відображати інтернет контент але і захищати користувача він небажаного відслідковування. Перевіримо функціональність зміни IP-адрес при користуванні сервісом. Хост на якому знаходиться web-сервіс має динамічні IP-адреси тому адреси можуть змінюватися, що ще більше покращує функцію приховування особистості. Для перевірки IP-адрес та фізичного розташування скористаємося сервісом whatsmyip.com (Рис.А.6).

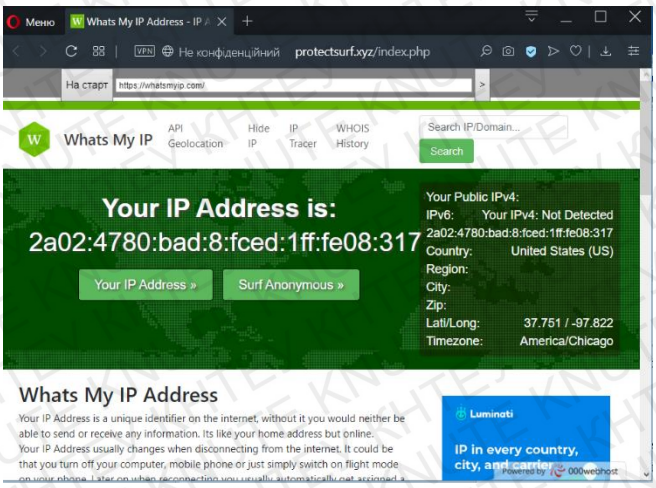


Рис.А.6. Перевірка IP-адреси.

Різні сервери мають використовувати різні IP-адреси відповідно (Рис.А.7).

Рис. А.7, аркуш 1. Перевірка IP-адреси на різних серверах.

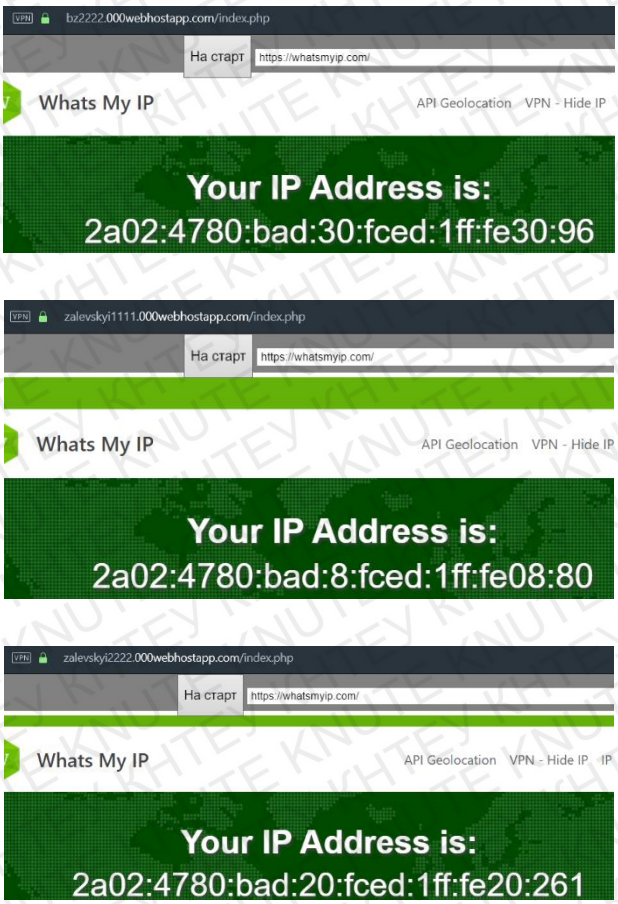


Рис. А.7, аркуш 2. Перевірка IP-адреси на різних серверах.

Мобільна версія теж була протестована. Для портативних пристроїв такі міри захисту мають бути доступними, оскільки саме такі пристрої найчастіше користуються незахищеними з'єднаннями (Рис. А.8).

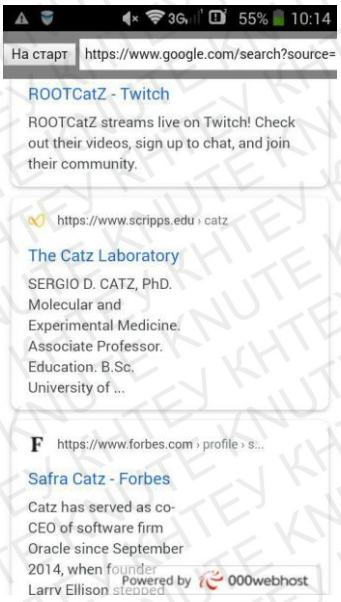


Рис. А.8. Відображення сервісу на смартфоні.

Також можливе збереження готового підключення у вигляді закладок браузеру. Це означає що можна зберігати посилання відвідуваного ресурсу через сервіс та пізніше знову продовжити роботу з безпечним з'єднанням, не потребуючи підключення з самого початку (Рис.А.9).

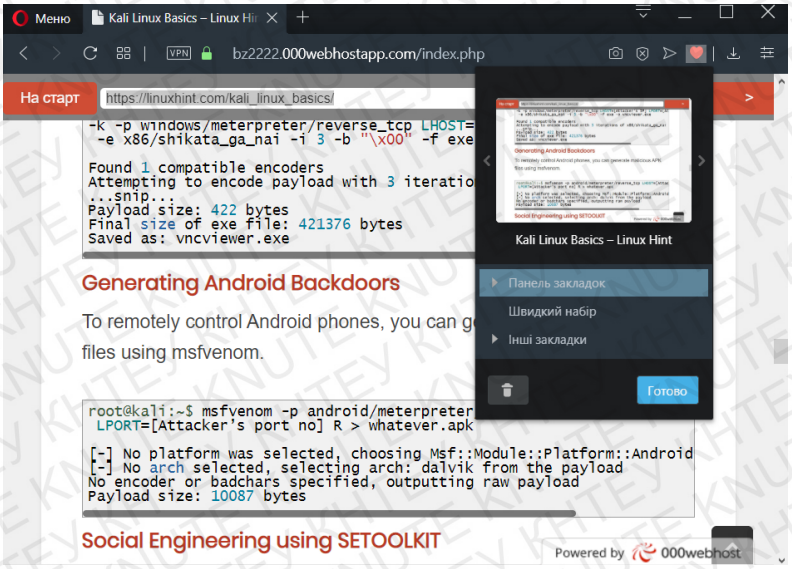


Рис. А.9. Взаємодія з функціоналом браузеру.

Додаток Б

КЕРІВНИЦТВО КОРИСТУВАЧА

Для початку роботи перейдіть за адресою protectsurf.xyz.

Далі оберіть один із існуючих серверів або продовжіть роботу в початковому (Рис.Б.1).

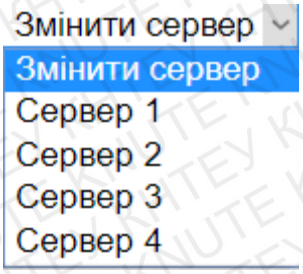


Рис.Б.1. Вибір серверу.

Для ручного приєднання до певного серверу, можна використати наступні посилання:

- https://zalevskyi1111.000webhostapp.com
- https://zalevskyi2222.000webhostapp.com
- https://bz1111.000webhostapp.com
- https://bz2222.000webhostapp.com

Уведіть або скопіюйте необхідну для відвідування адресу в поле введення (Рис.Б.2).

Рис.Б.2. Поле введення адреси.

Натисніть «Почати» для переходу на уведену сторінку. У випадку недоступності, над полем з’явиться повідомлення про помилку з відповідним HTTP кодом. У випадку помилки перевірте правильність написання адреси та оновіть поточну сторінку. Для зручного інтернет-серфінгу рекомендовано почати з пошукового серверу, тобто www.google.com або www.bing.com. Сервіс сумісний з сучасними браузерами, тому можна як зберігати сторінки в закладках, так і використовувати навігацію (Рис Б.3.).

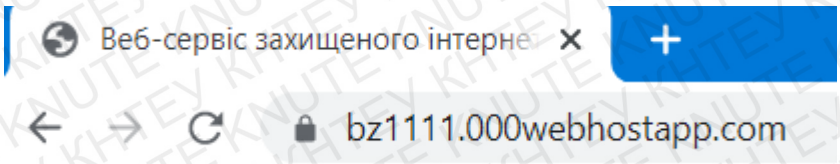


Рис. Б.3. Навігація браузеру

Сервіс сумісний з деякими розширеннями браузерів (наприклад Adblock, чи VPN), але не усі розширення можуть коректно працювати зі сторінками через web-сервіс, чи коректно відображати зміст цих сторінок (Рис.4).

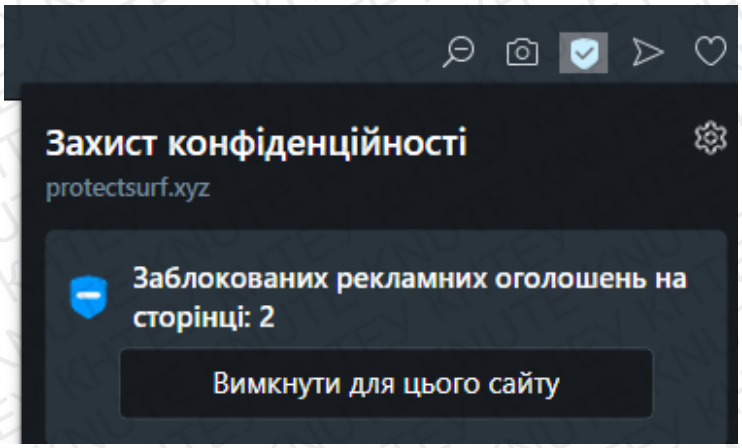


Рис. Б.4. Функціональність з використанням вбудованого блокувальника реклами

Після успішного приєднання, угорі сторінки з’явиться підменю швидкого доступу. З його допомогою можна змінити поточне посилання, скопіювати його, або повернутися на початкову сторінку поточного серверу. Для зміни сторінки, введіть нову адресу та натисніть піктограму «>» (Рис. Б.5).

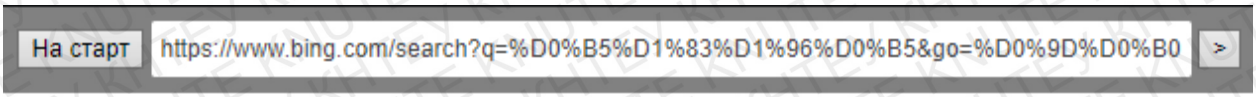


Рис. Б.5. Верхнє поле навігації, з функціоналом повернення на стартову стоінку та зміни адреси.

Для повернення на початкову сторінку, натисніть «На старт».

Додаток В

Лістинг основної бібліотеки Proxy.php що зв’язує файли

```
<?php
namespace Proxy;
//Налаштування використання наступних файлів бібліотеки
use Proxy\Event\ProxyEvent;
use Proxy\Http\Request;
use Proxy\Http\Response;
use Proxy\Config;
//Оголошення класу Proxy який і буде з'єднувати та надавати запити
class Proxy {
	private $dispatcher;
	private $request;
	private $response;
	private $output_buffering = true;
	private $output_buffer = '';
	private $status_found = false;
	public function __construct(){
	}
	public function setOutputBuffering($output_buffering){
		$this->output_buffering = $output_buffering;
```

```
        }

        private function header_callback($ch, $headers){

                $parts = explode(":", $headers, 2);

                // Вивантаження коду статусу

                if(preg_match('/HTTP\/1.\d+ (\d+)/', $headers, $matches) &&
stripos($headers, '200 Connection established') === false){

                        $this->response->setStatusCode($matches[1]);

                        $this->status_found = true;

                } else if(count($parts) == 2){

                        $name = strtolower($parts[0]);

                        $value = trim($parts[1]);

                        $this->response->headers->set($name, $value, false);

                } else if($this->status_found){

                // Запит до ProxyEvent

                        $event = new ProxyEvent(array('request' => $this->request,
'response' => $this->response, 'proxy' => &$this));

                        $this->dispatch('request.sent', $event);

                }

                return strlen($headers);

        }

        private function write_callback($ch, $str){

                $len = strlen($str);
```

```
        $this->dispatch('curl.callback.write', new ProxyEvent(array(
            'request' => $this->request,
            'data' => $str
        )));
        if($this->output_buffering){
            $this->output_buffer .= $str;
        }
        return $len;
    }
    private $listeners = array();
    // Визначення proxy як відправника
    public function getEventDispatcher(){
        return $this;
    }
    public function addListener($event, $callback, $priority = 0){
        $this->listeners[$event][$priority][] = $callback;
    }
    public function addSubscriber($subscriber){
        if(method_exists($subscriber, 'subscribe')){
            $subscriber->subscribe($this);
        }
```

```
    }
    //Обробка направлених подій
    private function dispatch($event_name, $event){
        if(isset($this->listeners[$event_name])){
            $temp = (array)$this->listeners[$event_name];
            foreach($temp as $priority => $listeners){
                foreach( (array)$listeners as $listener){
                    if(is_callable($listener) ){
                        $listener($event);
                    }
                }
            }
        }
    }
    public function forward(Request $request, $url){
        // Запит на зміну URL коду через request
        $request->setUrl($url);
        // Підготовка Запиту та відповідей
        $this->request = $request;
        $this->response = new Response();
        $options = array(
```

```
                CURLOPT_CONNECTTIMEOUT    => 10,
                CURLOPT_TIMEOUT           => 0,
            );
// Перевірка на curl та отримання даних
            $config_options = Config::get('curl', array());
            $options = array_merge_custom($options, $config_options);
            $options[CURLOPT_HEADERFUNCTION] = array($this,
'header_callback');
            $options[CURLOPT_WRITEFUNCTION] = array($this,
'write_callback');
// Конфігурація обробника надсилання та отримання
            $this->dispatch('request.before_send', new ProxyEvent(array(
                'request' => $this->request,
                'response' => $this->response
            )));
?>
```

Лістинг коду зв’язуючого файлу index.php

```
<?php
define('PROXY_START', microtime(true));
// Підключення рушія, звернення до autoload для перевірки файлів
require("engine/autoload.php");
use Proxy\Http\Request;
```

```
use Proxy\Http\Response;
use Proxy\Config;
use Proxy\Proxy;
// Початок сесії
session_start();
// Завантаження файлу конфігурації, перевірка ключів
Config::load('./config.php');
if(!Config::get('app_key')){
	die("app_key необхідний в config.php");
}
if(!function_exists('curl_version')){
	die("cURL не завантажено");
}
//Визначення способу генерування URL адрес
if(Config::get('url_mode') == 1){
	Config::set('encryption_key',
md5(Config::get('app_key').$_SERVER['REMOTE_ADDR']));
} else if(Config::get('url_mode') == 2){
	Config::set('encryption_key', md5(Config::get('app_key').session_id()));
}
session_write_close();
```

```
// Очікування підтвердження
if(isset($_POST['url'])){
        $url = $_POST['url'];
        $url = add_http($url);
        header("HTTP/1.1 302 Found");
        header('Location: '.proxify_url($url));
        exit;
} else if(!isset($_GET['q'])){
//Перенаправлення в разі відсутності
        if(Config::get('index_redirect')){
                header("HTTP/1.1 302 Found");
                header("Призначення: ".Config::get('index_redirect'));
        } else {
                echo render_template("./forms/main.php", array('version' =>
Proxy::VERSION));
        }
        exit;
}
// Декодування параметрів для доступу до url
$url = url_decrypt($_GET['q']);
$proxy = new Proxy();
```

```
// Завантаження плагінів;
foreach(Config::get('plugins', array()) as $plugin){
    $plugin_class = $plugin.'Plugin';
    if(file_exists('./plugins/'.$plugin_class.'.php')){
// Використовування плагінів з /plugins/
        require_once('./plugins/'.$plugin_class.'.php');
    } else if(class_exists('\\Proxy\\Plugin\\'.$plugin_class)){
//Перевірка плагінів
        $plugin_class = '\\Proxy\\Plugin\\'.$plugin_class;
    }
    // Відправлення запиту на index.php
    $request = Request::createFromGlobals();
    // Очистка GET параметрів
    $request->get->clear();
// Запит пересилання на інший URL
    $response = $proxy->forward($request, $url);
    $response->send();
} catch (Exception $ex){
// Повернення на випадок помилки
    if(Config::get("error_redirect")){
        $url = render_string(Config::get("error_redirect"), array(
```

```
                'error_msg' => rawurlencode($ex->getMessage())
            ));
// Повідомлення що заголовки вже надіслані
            header("HTTP/1.1 302 Found");
            header("Призначення: {$url}");
        } else {
//Звернення та відображення форми введення
            echo render_template("./forms/main.php", array(
                'url' => $url,
                'error_msg' => $ex->getMessage(),
                'version' => Proxy::VERSION
            ));
        }
}
?>
```

Лістинг коду конфігурації config.php

```
<?php
$config = array();
// Ключ ідентифікації процесу, потрібно ввести вручну
$config['app_key'] = 'bohdantest';
// Ключ для base64 шифрування, потрібно ввести вручну
$config['encryption_key'] = 'bohdantest';
//Налаштування унікальності url адрес: 1 - жодного шифрування
2 - Base64 кодування, з можливістю копіювання та збереження URL
$config['url_mode'] = 2;
// Список плагінів для завантаження по порядку
$config['plugins'] = array(
        'HeaderRewrite',
        'Stream',
// Стандартні необхідні для роботи плагіни рушія
        'Cookie',
        'Proxify',
        'UrlForm',
// Плагіни для спецефічних web-ресурсів
        'Youtube',
); return $config; ?>
```

Лістинг коду титульної сторінки main.php

```
//Графічне оформлення сторінки
<!DOCTYPE html>
<html>
<head>
<title>web-сервіс захищеного інтернет серфінгу</title>
<style type="text/css">
html body {
		font-family: Arial,Helvetica,sans-serif;
		font-size: 12px;
		background-image: url("back.png");
}
#container {
		width:500px;
		margin:0 auto;
		margin-top:150px;
}
#error {
		color:red;
		font-weight:bold;
```

```
}
#frm {
        padding:15px 15px;
        background-color:#818181;
        border:1px solid #818181;
        border-radius: 12px;
}
#opuc {
        text-align:center;
        font-size:10px;
        margin-top:10px;
        clear:both;
}
</style>
</head>
<body>
<div id="container">
<div style="text-align:center;">
<h1 style="color:blue;">Введіть адресу для доступу</h1>
</div>
<?php if(isset($error_msg)){ ?>
```

```
        <div id="error">
        <p><?php echo strip_tags($error_msg); ?></p>
        </div>
//Код прив’язки уведених даних до index.php звідки і передається у проксі
        <?php } ?>
        <div id="frm">
<form action="index.php" method="post" style="margin-bottom:0;">
<input name="url" type="text" style="width:400px;" autocomplete="off"
placeholder="https://" />
<input type="submit" value="Почати" />
</form>
        <script type="text/javascript">
        document.getElementsByName("url")[0].focus();
        </script>
        </div>
        </div>
        <div id="opuc">
                        <select
onchange="window.location=this.options[this.selectedIndex].value">
            <option value="">Змінити сервер</option>
            <option value="https://zalevskyi1111.000webhostapp.com">Сервер
1</option>
```

```
        <option value="https://zalevskyi2222.000webhostapp.com">Сервер 2</option>
        <option value="https://bz1111.000webhostapp.com">Сервер 3</option>
        <option value="https://bz2222.000webhostapp.com">Сервер 4</option>
      </select>
    </div>
    <div id="opuc">
    Веб-сервіс створив <a href="mailto:bogdan1297@ukr.net" target="_blank">Залевський Б.П.</a>;
    Проксі рушій <a href="//github.com/Athlon1600/php-proxy" target="_blank">PHP-Proxy</a>;
    Веб-серверні технології <a href="https://httpd.apache.org" target="_blank">Apache</a></div>
    </body>
    </html>
```

Лістинг коду urLform.php

```
    <style type="text/css">
    html body {
    margin-top: 50px !important;
    }
    #top_form {
```

```
        position: fixed;
        top:0;
        left:0;
        width: 100%;
        margin:0;
        z-index: 2100000000;
        -moz-user-select: none;
        -khtml-user-select: none;
        -webkit-user-select: none;
        -o-user-select: none;
        border-bottom:1px solid #818181;
         background:#818181;
        height:45px; line-height:45px;
}
#top_form input[name=url] {
        width: 550px;
        height: 20px;
        padding: 5px;
        font: 13px "Helvetica Neue",Helvetica,Arial,sans-serif;
        border: 0px none;
        background: none repeat scroll 0% 0% #FFF;
```

```
}
</style>
<script>
var url_text_selected = false;
function smart_select(ele){
                ele.onblur = function(){
                    url_text_selected = false;
                };
                ele.onclick = function(){
                    if(url_text_selected == false){
                        this.focus();
                        this.select();
                        url_text_selected = true;
                    }
                };
}
</script>
<div id="top_form">
<div style="width:800px; margin:0 auto;">
<form method="post" action="index.php" target="_top" style="margin:0;
padding:0;">
```

```
<input type="button" value="На старт"
onclick="window.location.href='index.php'">
<input type="text" name="url" value="<?php echo $url; ?>"
autocomplete="off">
<input type="hidden" name="form" value="1">
<input type="submit" value=">">
</form>
</div>
</div>
<script type="text/javascript">
smart_select(document.getElementsByName("url")[0]);
</script>
```

Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

РЕФЕРАТ

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка та впровадження web-сервісу захищеного інтернет-серфінгу»

Студента 2 курсу, 5м групи, спеціальності 121 «Інженерія програмного забезпечення», спеціалізації «Інженерія програмного забезпечення» _____ підпис студента — Залевського Богдана Паловича

Науковий керівник кандидат технічних наук, професор _____ підпис керівника — Пашорін Валерій Іванович

Гарант освітньої програми доктор технічних наук, професор _____ підпис керівника — Криворучко Олена Володимирівна

КИЇВ – 2019

АНОТАЦІЯ

Залевський Б.П. Розробка та впровадження web-сервісу захищеного інтернет-серфінгу

У випускній кваліфікаційній роботі розглянуто розробку web-сервісу, його компонентів та функціональності, а також використання PHP, та технології Apache HTTP Server для побудови web-додатків. Також були розглянуті існуючі загрози та програмні методи захисту користувачів в інтернету.

Ключові слова: Розробка програмного забезпечення, програмне забезпечення, PHP, Apache, web-сервіс, проксі.

ANNOTATION

Zalevsky B.P. Development and implementation of a secure web surfing web service

In the final work examines the development of a web service, its components and functionality, as well as the use of PHP and Apache HTTP Server technology to build web applications. Existing threats and software methods for protecting users on the Internet were also considered.

Keywords: Software development, software, PHP, Apache, web service, proxy.

Загальна характеристика роботи

Актуальність обраної теми полягає в щоденному користування мережі інтернет понад мільярдом користувачів, що допомагає як в покращенні зв’язку і передачі даних, так і створенні проблем захисту персональних даних, конфіденційної інформації та анонімності в цілому.

Метою даної випускної кваліфікаційної роботи є розробка, тестування та впровадження web-сервісу захищеного інтернет серфігну, покращення навичок розробки web-сервісів.

Об’єктом дослідження є всесвітня мережа інтернет, проблематика захисту анонімності користувачів, загальні рекомендації організації захищеного інтернет з’єднання.

Методи дослідження: Інформаційні бази, авторські роботи, наукові журнали та періодичні видання, посібники з програмування та web-програмування, енциклопедичні дані, інтернет ресурси пов'язані з програмуванням.

Предметом дослідження є програмні засоби забезпечення додаткового захисту користувача, функціональність такого програного забезпечення, його структура та складові, web-орієнтовані мови програмування та методи проектування.

Основними завданнями випускної квалфікаційної роботи є:

- Здійснити аналіз актуальних загроз персональних даних та анонімності;
- Розробити загальні рекомендації мережевої гігієни;
- Аналіз та опис існуючих програмних засобів захисту в мережі;
- Проектування моделі захищеного інтернет-серфінгу;
- Розробка та впровадження web-сервісу захищеного інтернет серфінгу;
- Тестування розробки.

Наукова новизна полягає в розробці web-сервісу на базі доповнених, уточнених даних актуальних загроз та існуючого пз, з використанням сучасних мов програмування та технічної документації.

Методами дослідження є численні інформаційні бази та авторські роботи, журнали та періодичні вилання, посібники з web-програмування.

В ході виконання було досліджені існуючі загрози анонімності, та персональним даним користувача, а також інструментальні засоби боротьби з ними. Безпека інформації інтернет користувачів на сьогодні є чи не основним питанням що часто піднімається як серед повсякдених користувачів, так і серед науковців та державних діячів. Функціональність програмних інструментів довзоляє на високому рівні уберегти себе, а їх різноманітність дозволяє обирати, від персонального захисту до захисту цілих корпоративних внутрішніх мереж.

Програмні засоби та онлайн сервіси надають різні послуги, але мають спільну мету забезпечення захисту клієнтів. Проблематика покращення засобів захисту буде підніматися частіше з всебільшим розвитком технологій, та загроз користувачеві. В ході виконання було розроблено web-сервіс захищеного інтернет-серфінгу, що дозволяє вільно переглядати інтернет сторінки з безпекою для клієна.