Міністерство освіти і науки України

СОЦІАЛЬНА ІНЖЕНЕРІЯ

Методичні вказівки до практичних занять
для здобувачів першого (бакалаврського) рівня вищої освіти
галузь знань 12 (F) Інформаційні технології
денної та заочної форм навчання

Луцьк 2026

УДК 004.056 (07)
С59

Рекомендовано до видання вченою радою факультету КІТ ЛНТУ,
протокол № _____ від «_____» _____ 20___ року.

Голова вченої ради факультету КІТ _____ Інна КОНДІУС

Електронна копія друкованого видання передана для внесення в репозитарій ЛНТУ

Директор бібліотеки _____ Наталія ПОЛІЩУК

Розглянуто і схвалено на засіданні кафедри комп'ютерної інженерії та безпеки ЛНТУ, протокол № _____ від « _____» _____ 20____ року.

Завідувач кафедри КІБ _____ Тарас ТЕРЛЕЦЬКИЙ

Укладач: _____ Оксана МІСКЕВИЧ, старший викладач кафедри комп'ютерної інженерії та безпеки ЛНТУ

Рецензент: _____ Світлана ЛАВРЕНЧУК, кандидат технічних наук, доцент кафедри комп'ютерної інженерії та безпеки ЛНТУ

Відповідальний за випуск: _____ Тарас ТЕРЛЕЦЬКИЙ, кандидат технічних наук, доцент кафедри комп'ютерної інженерії та безпеки ЛНТУ

С59 Соціальна інженерія: методичні вказівки до практичних занять для здобувачів першого (бакалаврського) рівня вищої освіти галузь знань 12 (F) Інформаційні технології денної та заочної форм навчання / уклад. О. І. Міскевич. Луцьк: ЛНТУ, 2026. 75 с.

Методичні вказівки до практичних занять з дисципліни «Соціальна інженерія» складено відповідно до діючої програми курсу.

Призначені для здобувачів вищої освіти галузі знань 12 (F) Інформаційні технології.

ЗМІСТ

ВСТУП

Методичні вказівки призначені для практичних занять з дисципліни «Соціальна інженерія», які формують у студентів психологічні аспекти та уміння для успішної боротьби з кібератаками Основними завданнями є: вивчення та оволодіння основними концепціями, методами та напрямами соціальної інженерії.

У процесі виконання завдань студенти поглиблюють свої знання від видів соціальної інженерії до заходів протидії соціальній інженерії

«Теоретико-методологічні основи соціальної інженерії» охоплює види соціальної інженерії, психологію та основні схеми впливу, збір інформації за допомогою соціальних мереж та пошукових систем, підбір облікових даних. Студенти вивчають аналізувати, як і чому відбувається збір інформації через соціальні мережі та пошукові системи, користуватися OSINT-даними.

«Атаки соціального інженера та заходи протидії соціальній інженерії» присвячений поглибленому вивченню цілі атак соціальної інженерії: отримання конфіденційної інформації, отримання доступу до систем та ресурсів, маніпулювання користувачами. Вміти реагувати на інцидент та відновлення систем.

Таким чином, дані методичні вказівки забезпечують практичну підготовку здобувачів вищої освіти у сфері соціальної інженерії та сприяють формуванню знань та навичок.

Практична робота №1.
Використання можливостей Google Dorking для пошуку інформації у відкритому доступі

Мета: ознайомитись технікою Google Dorking. Провести оцінку безпеки згідно знайдених у відкритому доступі даних

Короткі теоретичні відомості

Користувачі інформаційних систем часто самі надають можливості соціальним інженерам щодо збору чутливої інформації. Це не лише персональні дані, а й інша конфіденційна інформація, яка випадково, або з недосвідченості користувача потрапляє у відкритий доступ.

Google Dorks або Google Hacking – техніка, яка використовується соціальними інженерами для створення запитів в різних пошукових системах з метою виявлення прихованої інформації. Це метод, в якому звичайні запити на пошук веб-сайтів використовуються для одержання інформації, яка колись потрапила у мережу інтернет. Використовується техніка із застосуванням спеціального пошукового синтаксису. Додатково до чутливої інформації техніка дає можливість встановити версії застосунків, які використовуються в інформаційній системі, і відповідно дізнатись про властиві їм вразливості. Це не типовий застосунок для соціальної інженерії, але може використовуватись на доданок до одержаної інформації наприклад, для спостереження за цільовим об’єктом. Наприклад, рядок пошуку, заданий у виді inurl: «ViewerFrame? Mode =», дає змогу знайти загальнодоступні веб-камери. Також соціальний інженер може шукати загальнодоступні файли, що містять облікові дані користувачів, що працюють у цільовій компанії.

Оператором може бути ключове слово або фраза, наприклад: «inurl», «intext», «site», «feed», «language». За кожним оператором слідує двокрапка, за яким слід відповідний ключова фраза або фрази. Ці оператори дозволяють виконувати пошук більш конкретної інформації, наприклад: певні рядки тексту всередині сторінок веб-сайту або файли, розміщені по конкретному URL-адресою. Крім іншого, Google Dorking може також знаходити приховані сторінки для входу в систему, повідомлення про помилки, що видають інформації про доступні вразливості і файли загального

доступу. В основному причина полягає в тому, що адміністратор веб-сайту міг забути вилучити їх з відкритого доступу.

Дані, які можна розшукати із використанням Google Dorking:

– логіни та паролі;

– сторінки логіну для адміністраторів веб-сайту;

– документи, в тому числі й конфіденційні;

– Аадреси електронних скриньок;

– дані банківських рахунків;

– файли будь-яких форматів(pdf, excel, txt та інші);

– HTML заголовки із визначеним вмістом;

– журнали подій;

– властивості застосунків (версія, притаманні вразливості тощо);

– номери телефонів;

– камери відеонагляду з вільним доступом;

– фотографії.

Синтаксис, який використовується при цьому, в загальних рисах описано в таблиці 1.1.

Таблиця 1.1 – Основні правила синтаксису

Оператор	**Властивості**
inurl:	Включення [inurl:] до запиту дає можливості обмеження результатів до сторінок, що містять це слово в відповідному URL. Наприклад, [inurl:google search] поверне документи, у яких в URL-адресі згадується слово "google", і згадується слово «search» у довільному місці сторінки. "inurl:" перед кожним словом у запиті - еквівалентне "allinurl:" перед запитом: [inurl: google inurl: search] еквівалентне [allinurl: google search].

Продовження таблиці 1.1

allinurl:	Якщо запит починається з [allinurl:], Google обмежить результати лише тими, що містять усі слова запиту в URL- адресі. Наприклад, [allinurl: google search] поверне лише ті документи, які містять і «google», і «пошук» у url. Оператор [allinurl:] працює над словами, а не компонентами url, та ігнорує знаки пунктуації. Таким чином, [allinurl: foo/bar] обмежить результати сторінкою зі словами «foo» і «bar» в URL-адресі, але не вимагатиме, щоб вони були розділені косою рискою всередині цієї URL-адреси, щоб вони були поруч, або щоб вони були в тому самому порядку розташування.
intitle:	Включення [intitle:] приводить до обмеження результатів пошуку сторінками, що містять це слово в заголовку. Наприклад, [intitle: google search] поверне сторінки, у яких у заголовку згадується слово «google», і слово «search» в будь-якому місці сторінки (заголовок чи ні).
allintitle:	Якщо запит розпочато з [allintitle:], Google обмежить результати лише тими, що містять усі слова запиту в заголовку. Наприклад, [allintitle: google search] поверне лише ті документи, у яких у заголовку є і «google», і «search».
site:	Включення оператору [site:] дає змогу обмежити результати лише тими веб-сайтами в даному домені. Наприклад, [help site: www.google.com] знайде сторінки про слово «help» серед www.google.com. [help site:com] знайде з згадуванням слова «help» за адресами .com.
stocks:	Початок пошуку з оператору [stocks:] призводить до обробки решти термінів запиту як символів біржі, і буде посилатися на сторінку, що показує біржову інформацію для цих символів. Наприклад, [stocks: intc yhoo] покаже інформацію про Intel та Yahoo. (Примітка: ви повинні вводити символи, а не назву компанії.)
define:	Запит [define:] надасть визначення слів, які ви вводите після нього, зібрані з різних Інтернет-джерел. Визначення буде стосуватися всієї введеної фрази (тобто, вона буде включати всі слова в тому порядку, в якому ви їх ввели).
info:	Запит [info:] надасть деяку інформацію про цю веб-сторінку, яку має Google. Зокрема, [info: www.google.com] відображатиме інформацію про домашню сторінку Google.

Продовження таблиці 1.1

related:	Запит [related:] веб-сторінки, які «схожі» на вказану веб- сторінку. Наприклад, [related: www.google.com] перелічить веб-сторінки, подібні до домашньої сторінки Google.
link:	Запит [link:] веб-сторінки, які мають посилання на вказану веб-сторінку. Наприклад, [посилання: www.google.com] перелічить веб-сторінки, посилання на яких спрямовані на домашню сторінку Google.
ache:	Якщо ви включите в запит інші слова, Google виділить ці слова в кешованому документі. Наприклад, [cache: www.google.com web] покаже кешований вміст із виділеним словом «web». Запит [cache:] покаже версію веб-сторінки, яку Google має у своєму кеші. Наприклад, [cache: www.google.com] покаже кеш Google домашньої сторінки Google.

Оператори можуть використовуватись у сукупності, що дає змогу розширити можливості та точність пошуку.

Наприклад, site:trello.com "/wp-admin" password

Техніка Google Dorking може використовуватись і для цілей безпеки організації, зокрема:

Для виявлення “цифрового сліду” організації та її персоналу в інтернет.

Висновки про доступні в інтернет дані, для передбачення потенційно можливих кібератак.

Виявлення вразливостей веб-сайтів та неприйнятного вмісту, що з’являється них.

Відстеження шкідливих дій, націлені на бізнес конкретної організації.

Файл robots.txt або індексний файл - звичайний текстовий документ в кодуванні UTF-8, який діє для протоколів http, https, а також FTP. Файл дає пошуковим роботам рекомендації: які сторінки / файли варто сканувати. Якщо файл буде містити символи не в UTF-8, а в іншому кодуванні, пошукові роботи можуть неправильно їх обробити. Правила, перераховані в чинному файлі robots.txt, призначені тільки щодо того хоста, протоколу і номера порту, де розміщений файл.

Індивідуальні завдання

1. Задатись необхідними вихідними даними: назвою організації та даними користувача/користувачів. (Дані, одержані у ході пошуку, можуть бути використані лише в навчальних цілях). В якості вихідних даних рекомендовано взяти організацію, в якій працює або навчається студент Вашої групи, його ПІБ та дані колег. Або якщо організація недостатньо представлена у інтернет-просторі, задатись даними будь-якої відомої компанії.

2. Здійснити пошук особистих сторінок 2 студентів Вашої групи у соціальних мережах, де згадано назву організації.

3. Використовуючи оператори «info» та «related», знайти список схожих веб сайтів та конкретну інформацію, представлену в заголовку сайту організації:

4. Використати оператори site:$organization.com; inurl:robots.txt; site:$organization.com: intitle:index.of; site:$organization.ua inurl:backup; intitle:index.of inurl:admin.

Зробити висновки щодо кожного з операторів, які дані вони можуть дати.

5. Виконати пошук корпоративної пошти студентів Вашої групи, наприклад використовуючи оператори.

6. Опробувати пошук інформації про паролі, з умовою не використовувати у шкідливих цілях видобуту інформацію, доповнити ці операції файлами інших типів, наприклад sql, doc та інші.

7. Задатись переліком ключових слів, та виконати пошук виду за різними типами файлів: $organization ext:xls ~конфіденційно; $organization budget filetype:xlsx OR budget filetype:xml;

8. Знайдіть усі документи Word з ключовим словом «звіт».

9. Знайдіть усі PDF-файли на сайті обраного університету/організації.

10. Здійснити пошук зображень: $organization filetype:img OR filetype:pn.

11. Складіть список відкритих директорій із доступом до файлів.

12. Визначте, чи є на обраному сайті форми авторизації (login).

13. Google Dorks, які треба використати у своїй роботі:

site: – пошук інформації на конкретному сайті.

intitle: – пошук веб-сторінок з певними словами у заголовку.

inurl: – пошук веб-сторінок з певними словами в URL-адресі.

filetype: – пошук файлів певного типу. і

ntext: – пошук веб-сторінок з певними словами в тексті.

link: – пошук сторінок, що посилаються на певну URL-адресу.

cache: – перегляд кешованої версії сторінки.

related: – пошук сторінок, схожих на вказану URL-адресу.

info: – отримання інформації про веб-сторінку.

define: – пошук визначень слів.

Тильда (~) та мінус (-) – пошук синонімів та виключення слів з пошуку.

AROUND(number) – пошук слів, розташованих поруч одне з одним на веб-сторінці.

Зміст звіту

1. Тема та мета практичного заняття.
2. Результати у виді скріншотів та поясненнями щодо одержаних результатів завдань 1-13.
3. Рекомендації, які можуть запобігти витоку чутливих даних у відкритий доступ.
4. Відповіді на контрольні запитання.

Контрольні запитання

1. Що таке Google Dorking і для чого він використовується?
2. Чим Google Dorking відрізняється від звичайного пошуку?
3. Які ризики пов’язані з використанням Google Dorking?
4. Як захистити сайт від індексації конфіденційних даних?
5. Якими є причини витоку чутливих даних у відкритий доступ?
6. Які дані пошуку за допомогою цієї техніки може використати соціальний інженер?

Література: [2, 3, 10, 11, 12, 14]

Практична робота №2.
Аналіз методів, які використовують шахраї в соціальній інженерії

Мета: ознайомити основними видами соціальної інженерії. Вивчити приклади реалізації атак різних типів. Ідентифікувати методи соціальної інженерії та виявити сайти вразливі до перехвату даних

Короткі теоретичні відомості

Основні техніки соціальної інженерії включають в себе:

1. Фішинг-атаки – цей вид шахрайства є найбільш поширений у соціальній інженерії. Фішингова атака полягає в незаконному одержані конфіденційних даних користувача. Це може бути логін і пароль. Дуже часто фішингові листи можуть містити граматичні помилки. В таких листах зловмисники надають гіперсилку на відповідну копію сайта (наприклад, поштового клієнта) з формою, в якій необхідно ввести свій логін, пароль та іншу особисту інформацію. Одним із прикладів фішингу є збір логінів і паролів користувачів, саме шляхом розсилання листів і повідомлень, які спонукають потенційну жертву повідомити необхідну інформацію. Щоб унебезпечити користувача від таких зловмисників, необхідно ігнорувати листи від невідомих адресатів.

2. Претекстинг – це така атака, яку проводять за завчасно підготовленим сценарієм. Такі атаки націлені на появу почуття довіри потенційної жертви до зловмисника. Такі атаки зазвичай здійснюють по телефону. Такий метод зазвичай не вимагає від зловмисника попередньої підготовки і щодо пошуку даних про жертви. Основна ідея претекстинг полягає у видачі себе за іншу людину з метою одержання бажаних даних.

Джерела відкритого доступу є способом одержати інформацію про людину. Зазвичай в основному – це сторінки соціальних мереж.

3. Троянський кінь. Ця техніка заснована на якості цікавості жадібності потенційної жертви. Соціальний інженер може відправити електронний лист, який містить безкоштовне відео або оновленням будь-якої програми у вкладенні. Потенційна жертва зберігає ці файли, які є троянськими програмами. Така техніка буде залишатися ефективною до того часу, поки відповідні користувачі будуть

бездумно зберігати або відкривати будь-які вкладення.

4. Квипрокво. Під час застосування такого виду атаки зловмисники можуть обіцяти жертві якусь вигоду в обмін на факти. Зазвичай зловмисник може подзвонити в будь-яку компанію та відрекомендуватися співробітником IT-компанії і запропонувати встановити «необхідне» програмне забезпечення. Як тільки зловмисник отримає згоду на виконання такої роботи, порушник може отримати доступ як до системи, так і до усіх даних, що зберігаються в ній.

5. Tailgating (зворотний зв'язок) – це несанкціонований прохід на територію зловмисника разом із користувачем, який має права на доступ через пропускний пункт.

6. Плечовий серфінг. Такий вид застосовують у різноманітних громадських місцях. Це дозволяє зловмиснику спостерігати за комп'ютерними пристроями і телефонами через плече потенційної жертви. Інколи є ситуації, коли користувач сам пропонує зловмиснику потрібну інформацію, думаючи про порядність людини. У такому разі можна говорити про зворотну соціальну інженерію.

7. Служби миттєвого обміну повідомленнями. Сьогодні всі користувачі використовують обмін повідомленнями в режимі реального часу за допомогою мереж Viber, Telegam та ін. Доступність і швидкість такого способу спілкування робить такі служби відкритими для різноманітних атак. Як рекомендація щодо безпеки краще ігнорувати повідомлення від невідомих користувачів, а також не повідомляти їм особисту інформацію, не переходити за надісланими посиланнями.

Відомий спеціаліст з кібербезпеки Патрик Ф. Уілбур (Patrick F. Wilbur) проводив експеримент із перехоплення публічного трафіку Wi-Fi. Патрик Ф. Уілбур організував виключно з метою оцінювання рівня безпеки в інтернеті. Він перехоплював та аналізував трафік публічної мережі Wi-Fi впродовж кількох годин. І ніхто не лише не подзвонив у поліцію, а навіть не звернув на нього увагу. Люди заходили на свої улюблені сайти, такі як Netflix та Google, через протокол HTTP, здійснювали телефонні дзвінки й взагалі надсилали через інтернет купу нешифрованого трафіку, який можна було перехопити та модифікувати для подальших фішингових або vishing-атак.

Багато хто неправильно вважає, що увесь веб-трафік шифрується і тому можна спокійно пересилати конфіденційні дані навіть у публічних місцях. На жаль, експеримент призвів до вельми негативних висновків: величезний обсяг онлайн-трафіку є сьогодні абсолютно незашифрованим й це залишає великий ризик кібератаки.

У межах свого експерименту Патрик Уілбур установив SSID-ідентифікатор «Вільний гостьовий Wi-Fi». Така назва мережі є досить популярною та прийнятною. Під час під'єднання автоматично відкривалася pop-up-сторінка, що містила небагатослівну угоду про те, що користувач повинен погодитися з тим, що в мережі будуть відстежуватися його дані та комунікації.

Загалом, це був дуже дружній спосіб проведення експерименту. Справжній хакер був би набагато агресивнішим.

Щоб додатково захистити конфіденційність користувачів, Патрик Уілбур написав невелику програму для збирання статистичних даних про протоколи та порти, що застосовували в мережі програмними додатками кінцевих споживачів.

Ця програма не записує жодних IP-адрес, MAC-адрес, імен хостів або інформації додатків і не може бути налаштована таким чином, щоб зробити це. Вона призначена лише для однієї мети – узагальнити типи пакетів і портів, що використовуються, найменш інтрузивним способом.

Як виявилося, є дуже багато охочих під'єднатися до відкритої мережі Wi-Fi. Впродовж одного дня було отримано таку статистику:

– усього під'єднано 49 пристроїв;

– усі 100 % прийняли умови у pop-up-вікні та надіслали дані;

– нуль пристроїв використовували VPN.

Дехто помітить, що в цю статистику увійшли лише ті особи, які свідомо вибрали відкриту мережу та поставили відповідну позначку на pop-up-сторінці. Та люди, які обирають публічні мережі, більш імовірно здатні виконувати ризиковані дії в інтернеті. До речі, через те, що для під'єднання до мережі потрібно було підтвердити угоду на pop-up-сторінці, це виключало зі статистики будь-які автоматичні пристрої типу Internet of Things (IoT).

На жаль, HTTPS недостатньо для повного захисту. Насправді, ця технологія неправильно реалізована навіть на великих сайтах, які всім добре відомі та яким ви довіряєте.

Крім того, близько 42 % всього трафіку, що пройшов через мережу через вищезгаданий «гостьовий Wi-Fi», був нешифрованим HTTP-трафіком.

Після збору 489330 IP-пакетів виявилося, що:

– понад 42 % трафіку від загального обсягу на 80-му порту (Port 80) відносився до незашифрованого HTTP (проти майже 57 % на Port 443, що використовується протоколом HTTPS);

– 2638 пакетів — не шифровані пакети DNS;

– 18 пакетів належили до нешифрованих пакетів NTP- протоколу.

Оскільки протоколи DNS та NTP є небезпечними, а 42 % трафіку – це потенційно незашифрований трафік HTTP, що надсилається через порт 80, така статистика справді дуже турбує. А як щодо політик HTTP Strict Transport Security (HSTS), які повинні застосовувати веб-браузери?

Якщо вивчити поведінку декількох популярних веб-сайтів, то виявиться, що:

– популярні веб-сайти не завжди впроваджують HTTPS належним чином, якщо взагалі впроваджують (це містить, зокрема Google та Netflix);

– користувачі загальнодоступних мереж Wi-Fi залишаються вразливими до атаки MITM (man-in-the-middle), перехоплення приватних даних та інших атак.

Крім того, є й альтернативна статистика. Google використовує анонімну звітність користувачів Chrome, щоб виявити частоту застосування HTTPS в інтернеті.

Відповідно до власного звіту Google (станом на 29 грудня 2021 року):

– 11–31 % усіх веб-сайтів відвідуються без шифрування (доступ через незашифрований HTTP);

– ~ 7 % трафіку до сервісів Google не шифрується (навіть до 10 % для деяких продуктів Google);

– 82,6 % цього трафіку походить із мобільних пристроїв (що змушує з великим скептицизмом дивитися на використання мобільної ОС, розробленої Google).

Базова безпека в інтернеті останнім часом значно покращилася, але все ж таки недостатньо. Досі існують величезні проблеми, які не вдалося вирішити. Тому в публічних мережах Wi-Fi навіть сьогодні зловмисник може:

– дізнатися, які сайти ви відвідуєте (просто перехопивши запити DNS);

– здійснити атаки типу MITM (людина посередині) в момент завантаження сторінки HTTP;

– обмежити запобіжні можливості HSTS, вводячи фальшивий майбутній час через сервіс NTP (адже політики HSTS мають обмежений час дії);

– виконати фішинг-атаку для збирання конфіденційної інформації;

– провести телефонну атаку типу vishing на вас, ваших друзів чи вашу родину;

– ввести фальшивий вміст/рекламу або навіть майнити криптовалюту, використовуючи ваш процесор;

– обдурити вас та спонукати працювати з небезпечними плагінами, наприклад, застарілою версією Flash.

Індивідуальні завдання

1. Провести аналіз власних смс-повідомлень, повідомлень або дзвінків на наявність шахрайства. Дізнатися у родичів чи друзів, чи дзвонили або писали їм шахраї. Про що саме йшла мова (заблоковані платіжні картки, виграш автомобіля або великої грошової суми тощо). Визначити в який переважно час були дзвінки від шахраїв. Якщо було повідомлення або дзвінок від шахрая про заблоковану платіжну картку, чи повідомляли ви чи ваші близькі про шахрая банку яким користуєтесь. Виявити шаблони, які використовують шахраї. Зробити звіт.

2. Проаналізувати сайти, якими ви користуєтесь. Визначити сайти, які мають захищений протокол https та ті, які не мають захищений протокол https. Чи всі сайти, на яких ви проводите онлайн-платежі (покупки в інтернет-магазині, оплата комунальних чи інших платежів, користування електронними гаманцями, сервісами для поповнення електронних гаманців, банківських карток чи балансу мобільного зв’язку, сервісу переказу грошей тощо). Зробити звіт.

Зміст звіту

1. Тема та мета практичного заняття.
2. Результати у виді скріншотів або таблиць щодо одержаних результатів завдань.
3. Рекомендації, які можуть запобігти шахраям.
4. Відповіді на контрольні запитання.

Контрольні запитання

1. Які основні методи використовують шахраї в соціальній інженерії?
2. У чому полягає небезпека вішингу та як його розпізнати?
3. Як можна протидіяти методам соціальної інженерії на рівні користувача?
4. Які заходи повинні впроваджувати організації для запобігання успішним атакам?
5. Які реальні приклади атак соціальної інженерії відомі за останні 3 роки?

Література: [2, 3, 10, 11, 12, 14]

Практична робота №3.
Види соціальної інженерії

Мета: навчитись розпізнавати як здійснюються атаки з використанням соціальної інженерії. Проводити аналіз URL-адрес хостів, робити попередні висновки щодо безпечності ресурсів мережі Інтернет

Короткі теоретичні відомості

Більшість методів соціальної інженерії не вимагають особливих технічних знань з боку зловмисників, а отже використовувати ці методи може будь-хто – від дрібних злодіїв до досвідчених кіберзлочинців.

Існує багато методик, які підпадають під загальний термін соціальної інженерії в галузі кібербезпеки. Серед найвідоміших методик – спам та фішинг.

Спам – це масове розсилання небажаних листів. Найчастіше спам – це лист електронної пошти, який надсилається одразу на велику кількість адрес, але він також може бути доставлений через миттєві повідомлення, SMS та соціальні мережі. Власне, спам не є соціальною інженерією, однак в деяких кампаніях використовуються його види, такі як фішинг, цілеспрямований фішинг (spearphishing), вішинг (vishing), смішинг (smishing), а також поширення шкідливих вкладень або посилань.

Фішинг – це форма кібератаки, під час якої злочинець намагається завойовувати довіру до жертви для виманювання конфіденційної інформації. Для отримання даних зловмисники також створюють відчуття терміновості або застосовують тактику залякування. Варто зазначити, що фішингові кампанії можуть бути націлені на велику кількість випадкових користувачів або конкретну особу чи групу.

Фішинг – це форма атаки з використання соціальної інженерії, в ході якої зловмисник, маскуючись під надійний суб’єкт, вимагає конфіденційну інформацію жертв.

Чи отримували ви коли-небудь електронне повідомлення нібито з банку чи іншого популярного онлайн-сервісу, який вимагав «підтвердити» дані облікового запису, номер кредитної картки чи іншу конфіденційну інформацію? Якщо так, ви вже знаєте, як виглядає фішинг-атака. Ціль фішингу – отримання цінних даних, які

можуть бути продані або використані для зловмисних цілей, таких як вимагання, викрадення грошей або особистих даних.

Походження терміну

Концепція фішингу вперше була описана в 1987 році в документі з конференції під назвою «Безпека системи: перспективи хакера». В документі описувалась техніка зловмисників, яка полягає в імітації авторитетних організацій або сервісів. Саме слово є омофоном англійського слова «Fishing» (рибалка), оскільки техніка використовує ту ж логіку «вилову».

Яку мету переслідує фішинг?

Фішинг існує впродовж багатьох років, за цей час кіберзлочинці розробили широкий спектр методів інфікування жертв.

Найчастіше зловмисники, які займаються фішингом видають себе за банки чи інші фінансові установи, щоб змусити жертву заповнити фальшиву форму та отримати дані облікових записів.

У минулому для виманювання даних користувачів кіберзлочинці часто використовували неправильно написані або оманливі доменні імена. Сьогодні зловмисники використовують більш складні методи, завдяки чому фальшиві сторінки дуже схожі на свої легітимні аналоги.

Викрадені дані жертв, зазвичай, використовуються для викрадення коштів з банківських рахунків або продаються в Інтернеті.

Подібні атаки здійснюються також через телефонні дзвінки (vishing) та SMS-повідомлення (smishing).

Цілеспрямований фішинг. Кіберзлочинці, які використовують цей метод, зазвичай, заздалегідь детально досліджують свою ціль. Це значно ускладнює ідентифікацію вмісту як шкідливого.

Як розпізнати фішинг? Електронне повідомлення може містити офіційні логотипи або інші ознаки авторитетної організації. Нижче наведено кілька підказок, які допоможуть виявити фішингове повідомлення.

Загальні або неофіційні привітання – листи без персоналізації (наприклад, «Шановний клієнт») та формальностей, має викликати підозри. Те ж саме відноситься до псевдо-персоналізації з використанням випадкових, підроблених посилань.

Запит на особисту інформацію – часто використовують кіберзлочинці, але банки, фінансові установи та більшість онлайн-сервісів намагаються цього уникати.

Некоректна граматика – орфографічні та друкарські помилки, а також незвичайні фрази часто можуть означати небезпеку (але відсутність помилок не є доказом легітимності).

Несподівані повідомлення – будь-який незапланований контакт з банком повинен викликати підозри.

Терміновість – фішингові повідомлення часто намагаються викликати відчуття терміновості дій, залишаючи жертвам менше часу на роздуми.

Пропозиція, від якої важко відмовитися – якщо лист занадто хороший, щоб бути правдою, він, напевно, є фішинговим.

Підозрілий домен – чи дійсно американський чи німецький банк надсилатиме електронний лист з китайського домену?

Як захиститися від фішингу?

Щоб уникнути подібних атак, звертайте увагу на описані вище ознаки, за допомогою яких можна виявити фішингові повідомлення.

Дізнавайтесь про нові методи фішингу: читайте засоби масової інформації для отримання нової інформації про фішингові атаки, оскільки кіберзлочинці постійно знаходять нові методи для виманювання даних користувачів.

Не надсилайте облікові дані: будьте особливо уважні, коли у електронному листі начебто перевірені організації запитують ваші облікові або інші конфіденційні дані. У разі необхідності перевірте вміст повідомлення, відправника або організацію, яку вони представляють.

Не натискайте на підозрілі кнопки та посилання: якщо підозріле повідомлення містить посилання або вкладення, не натискайте та не завантажуйте вміст. Це може призвести до переходу на шкідливий веб-сайт або інфікувати ваш пристрій.

Регулярно перевіряйте облікові записи: навіть якщо ви не маєте підозр, що хтось намагається викрасти ваші облікові дані, перевірте банківські та інші облікові записи в Інтернеті на наявність підозрілої активності.

Інші методи зловмисників:

Цілеспрямований фішинг – це форма фішингу, під час якої зловмисник надсилає повідомлення,спрямовані на конкретну групу людей, або навіть просто окрему особу з метою викрадення даних або маніпулювання ними в зловмисних цілях.

Вішинг та смішинг – це методи соціальної інженерії, подібні до фішингу, але здійснюються не через електронну пошту. Зокрема, вішинг реалізовується через шахрайські телефонні дзвінки, а для смішингу використовуються текстові SMS-повідомлення, які містять шкідливі посилання або вміст.

Видавання себе за іншу особу є іншим популярним методом соціальної інженерії, під час якого кіберзлочинці діють нібито від імені певної особи, вводячи в оману потенційних жертв. Типовим прикладом є зловмисник, який видає себе за генерального директора певної компанії, укладає та затверджує шахрайські угоди в той час, як справжній генеральний директор перебуває у відпустці.

Афери з технічною підтримкою – це, зазвичай, неправдиві телефонні дзвінки або Інтернет-реклама, в якій зловмисники пропонують жертвам послуги служби технічної підтримки. Насправді, кіберзлочинці просто намагаються заробити гроші, продаючи фейкові послуги або усуваючи насправді неіснуючі проблеми.

Шкідливе програмне забезпечення, ціль якого викликати у жертви почуття страху чи тривоги та таким чином змусити її встановити небезпечний код на пристрій. Поширеними є випадки, коли користувачам відображалось повідомлення про нібито інфікування пристрою загрозою, для видалення якої необхідно завантажити антивірус (який, насправді, є шкідливим програмним забезпеченням).

Кібершахрайство – це схеми зловмисників, у яких часто використовують один або навіть декілька методів соціальної інженерії, описаних у цьому розділі.

Чому компанії малого та середнього бізнесу повинні боятися соціальної інженерії?

Відповідно до опитування, проведеного Zogby Analytics для Національного альянсу з кібербезпеки США у 2019 році, все більше компаній усвідомлюють, що вони є потенційними цілями кіберзлочинців. Зокрема, майже половина (44%) компаній з числельністю працівників 251-500 заявили, що стикалися з випадками втрати даних протягом останніх 12 місяців. Опитування також показало, що 88%

представників малого бізнесу вважають, що вони принаймні є «ймовірною» ціллю для кіберзлочинців, у тому числі майже половина (46%), вважають, що вони є «дуже ймовірною» ціллю.

За оцінками Центр прийому скарг на шахрайство в Інтернеті (IC3) при ФБР, лише в 2018 році внаслідок кібератак американські компанії втратили понад 2,7 мільярда доларів, в тому числі 1,2 мільярди доларів внаслідок атак з використанням компрометації ділової переписки(BEC)/компрометації електронних листів (EAC), які дозволяли несанкціоновано переказувати кошти.

Як здійснюються атаки з використанням соціальної інженерії?

Існує декілька ознак, які допоможуть ідентифікувати таку атаку. Зокрема, одна з них – погана граматика та правопис. Ще однією помітною ознакою є почуття терміновості, яке зловмисники намагаються створити для зменшення пильності жертви. Будь-який запит щодо конфіденційних даних також має викликати підозру: авторитетні компанії ніколи не просять відправити їм паролі або інші особисті дані електронною поштою або текстовими повідомленнями.

Деякі з ознак, які допоможуть виявити соціальну інженерію:

1. Погана граматика або занадто офіційна лексика.
2. Дивна адреса відправника.
3. Почуття терміновості.
4. Запит на конфіденційну інформацію.
5. Щось звучить занадто добре, щоб бути правдою.

П’ять способів захистити свою організацію від атак з використанням соціальної інженерії:

1. Регулярне навчання з кібербезпеки усіх працівників, включно з топ-менеджментом та IT-спеціалістами. Таке навчання повинно показувати та моделювати випадки з реального життя, оскільки методи соціальної інженерії розраховані на користувачів з низьким рівнем обізнаності у кібербезпеці.

2. Здійснюйте сканування на наявність слабких паролів, які потенційно можуть використати зловмисники для потрапляння до мережі вашої організації. Крім того, створіть додатковий рівень захисту за допомогою двофакторної аутентифікації.

3. Впроваджуйте рішення для захисту, які попереджають про можливі випадки шахрайства, а також повідомляють про виявлення спаму та фішингу.

4. Створіть політику безпеки з чітким планом дій, які потрібно буде виконати працівникам, якщо вони стикнуться з проявами соціальної інженерії.

5. Використовуйте рішення для централізованого управління корпоративною мережею, наприклад, ESET Security Management Center, щоб забезпечити повний огляд мережі, усіх рішень з безпеки та подій для виявлення та знешкодження потенційних загроз.

Відомі приклади

Систематичні фішинг-атаки почалися в мережі America Online (AOL) в 1995. Щоб викрасти легітимні облікові дані, зловмисники зв'язувалися з жертвами через AOL Instant Messenger (AIM), видаючи себе за співробітників AOL, які перевіряють паролі користувачів. Термін «фішинг» з'явився в групі новин Usenet, яка зосереджувалася на інструменті AOHell, який автоматизував цей метод, і так ім'я закріпилося. Після того, як AOL в 1997 році ввела контрзаходи, кіберзлочинці зрозуміли, що можуть використовувати таку ж техніку в інших галузях, зокрема й фінансових установах.

Одна з перших великих, хоча і невдалих, спроб була в 2001 році. Зловмисники, скориставшись хаосом від терористичних атак 9/11, розіслали потерпілим електронну розсилку нібито для перевірки посвідчення особи. Отримані дані використовувались для крадіжки банківських даних.

Вже у 2005 році за допомогою фішингу кіберзлочинці викрали у користувачів США понад 900 мільйонів доларів США.

Відповідно до дослідження глобального фішингу APWG, у 2016 році спостерігалося понад 250 тисяч унікальних фішингових атак, під час яких використовувалось рекордне число доменних імен, зареєстрованих зловмисниками, перевищуючи позначку в 95 тисяч. В останні роки кіберзлочинці намагалися зосередитися на банківських та фінансових послугах, користувачах електронного банкінгу, соціальних мереж, а також облікових даних електронної пошти.

Індивідуальні завдання

1. Вибрати 20 будь-яких онлайнових фішингових URL-адрес із відомого вам сервісу та знайти і зберегти інформацію про них: ім’я реєстратора; «час життя» домену (= дата закриття - дата реєстрації); країну IP-адреси.

2. Вибрати 20 будь-яких «чистих» URL-адрес, та знайти і зберегти про них інформацію .

3. Звести інформацію про фішингові і «чисті» WEB-ресурси до таблиці. Скласти таблиці частот для кожного із параметрів (реєстратор, час життя, країна) для чистих і фішингових URL.

4. За даними таблиць частот побудувати діаграми для кожного із параметрів.

5. Зробити висновки щодо можливих методів виявлення фішингових URL-адрес, та методів протидії загрозам фішингу.

Зміст звіту

1. Тема та мета практичного заняття.

2. Результати у виді скріншотів або таблиць щодо одержаних результатів завдань.

3. Висновки щодо можливих методів виявлення фішингових URL-адрес.

4. Відповіді на контрольні запитання.

Контрольні запитання

5. Які протоколи можуть бути вказані в URL і що вони означають з точки зору безпеки?

6. Як визначити доменне ім’я у структурі URL?

7. Чим відрізняються домени верхнього рівня (.com, .ua, .gov) з погляду довіри та безпечності?

8. Як за допомогою аналізу URL можна виявити фішингові сайти?

9. Що означає наявність/відсутність HTTPS у веб-адресі?

10. Яку роль відіграють параметри в URL і як шахраї можуть їх використовувати?

11. Чому важливо перевіряти адресу перед тим, як вводити особисті дані?

12. Які інструменти та сервіси можна використати для перевірки безпечності URL (VirusTotal, Whois, URLVoid)?

13. Які ризики пов’язані з переходом за скороченими посиланнями (bit.ly, t.co тощо)?

14. Як за допомогою аналізу SSL-сертифіката можна оцінити надійність сайту?

15. Які відмінності між офіційними та «дзеркальними» сайтами, і як це відобразиться в URL?

Література: [2, 3, 10, 11, 12, 14]

Практична робота №4.
Психологія та основні схеми впливу соціальної інженерії

Мета: ознайомитися з основними схемами психологічного впливу на людину. Пошук та ліквідації веб-шеллів. Протидія загрозі інсайдера

Короткі теоретичні відомості

Що ж таке «веб-шел» (web-shell)? Це якийсь шкідливий скрипт (програма), який зловмисники використовують для управління чужими сайтами і серверами: виконання команд терміналу, перебору паролів, доступу до файлової системи і т.п. Для розміщення скрипта найчастіше використовуються уразливості в коді сайту або підбір паролів.

Веб-шел являє собою серйозну загрозу насамперед для безпеки майданчика вашого сайту, адже розмістивши шел зловмисник отримує доступ до файлової системи і баз даних майданчика.

Хакери частіше за все, ставлять перед собою наступні завдання, коли ламають веб-сайт:

1. Дефейс сайту – частіше за все це робиться з метою самовираження (хактивізму) і часто злам не має продовження.

2. Вірусне зараження сторінок сайту – віруси додаються в кінець кожної сторінки і можуть бути досить різноманітної направленості. Головне завдання – доставити та виконати шкідливий код на ПК відвідувачів зламаного сайту

3. Розсилання спаму від імені сайту або пересилання (mail relay).

4. Зараження сайтів, що знаходяться по сусідству з зараженим і подальше їх зараження.

5. Завантаження веб-шелів і виконання довільних (часто, дуже небезпечних і протизаконних) дій, часто від імені власника зламаного веб-сайту.

6. Перетворення серверу веб-сайту на бота, який керується хакером з серверу контролю та управління (C&C) через протокол IRC і може виконувати довільні команди «хазяїна» (наприклад, брати участь у DoS-атаках на інші сайти).

7. Різновидів і завдань web-хаку досить багато, проте, характерними рисами майже усіх таких дій є:

– спроба подальшого розповсюдження ботів;

– непомітність слідів злому і зловмисних дій для справжніх власників сайтів, які зламано;

– корисливі інтереси (монетизація ботнету);

– використання вразливостей CMS веб.

Загальний порядок пошуку веб-шелів на сайті

Якщо наявність шелу очевидна необхідно:

– перевірте вміст файлів .htaccess веб-серверу;

– шукайте на сайті (або у папках всіх сайтів у випадку віртуального хостингу) підозрілі файли;

– коли файли знайдено, треба шукати всі файли, які схожі на підозрілі;

– дізнатись дату та час створення (зміни) цих файлів;

– пошукати іншу активність на сервері за цей час;

– якщо виявлено підозрілу активність — треба дізнатись (наприклад, з журнальних файлів веб-сервера) IP-адресу з якої вона здійснювалась;

– якщо знайдено IP-адресу джерела підозрілої активності — вивчити всі запити з цієї ж IP- адреси та визначити вразливість, яка була експлуатована та, можливо, додаткові вразливості веб-серверу;

– вжити заходів з вичищення зараження (змінених файлів тощо) та закриття вразливості;

– видалити все непотрібне для роботи сайту (старі копії сайту), кеш (він створюється заново), тимчасові файли.

Загальні поради з мінімізації шансів отримати шел на сайт

– Використовуйте тільки ті плагіни, що добре себе зарекомендували та необхідні для роботи сайту. Непотрібні плагіни треба відключати. Більшість сайтів зламуються внаслідок використання погано написаних (вразливих) плагінів.

– Перед завантаженням та інсталюванням плагіну подивіться скільки завантажень він має (часто, чим більше — тим краще). Перевірте, чи завантажуєте останню версію плагіну.

– Регулярно оновлюйте версію ядра сайту та плагіни. Більшість широко відомих ядер веб-сайтів нагадують про оновлення через адміністративну панель.

– Встановіть SSL-сертифікат та завжди його використовуйте для управління сайтом (приклад для WordPress - Administration over SSL).

– Цікавтесь підвищенням стійкості веб-сайту (приклади – Hardening WordPress, Security Checklist/Joomla! Setup, для інших ядер веб-сайтів — пошук в Інтернеті за словом «hardening»).

– Регулярно робіть резервні копії веб-сайту (наприклад, щотижневі та щомісячні копії)

Приклад шелу в державному секторі України

Приклад наводиться з метою допомоги у протидії веб-хаку системним адміністраторам та відповідальним за захист автоматизованих систем з підключенням до мережі Інтернет особам в владних структурах (та не тільки!).

Механізм виявлення, вивчення та знешкодження веб-шелу не є вичерпним та не гарантує 100% усунення вразливості, проте допоможе у більшості випадків уникнути компрометації веб-сайтів у найкоротші терміни та з мінімальною кількістю маніпуляцій. Передбачається, що Ви маєте доступ по SSH до веб-серверу під адміністративним обліковим записом. В Інтернеті можна знайти багато онлайн-сервісів з перевірки Вашого сайту на предмет ознак компрометації, наприклад проект http://sitecheck.sucuri.net, який показано на рисунку 4.1.

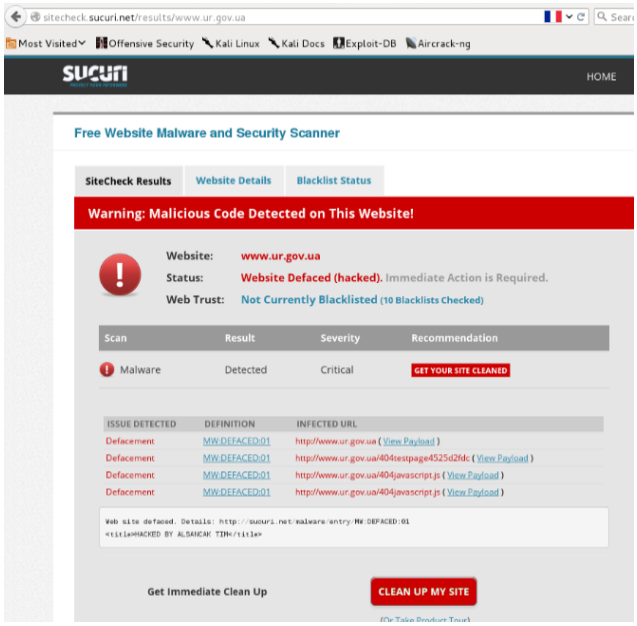

Рисунок 4.1 – Онлайн-сервісів з перевірки сайту

Загальний порядок пошуку веб-шелів на сайті:

– якщо наявність шелу очевидна — перевірте вміст файлів .htaccess веб-серверу;

– шукайте на сайті (або у папках всіх сайтів у випадку віртуального хостингу) підозрілі файли;

– коли файли знайдено, треба шукати всі файли, які схожі на підозрілі;

– дізнатись дату та час створення (зміни) цих файлів;

– пошукати іншу активність на сервері за цей час;

– якщо виявлено підозрілу активність – треба дізнатись (наприклад, з журнальних файлів веб-сервера) IP-адресу з якої вона здійснювалась;

– якщо знайдено IP-адресу джерела підозрілої активності – вивчити всі запити з цієї ж IP-адреси та визначити вразливість, яка біла експлуатована та, можливо, додаткові вразливості веб-серверу;

– вжити заходів з вичищення зараження (змінених файлів тощо) та закриття вразливості;

– видалити все непотрібне для роботи сайту (старі копії сайту), кеш (він створюється заново), тимчасові файли.

Знаходимо шел у домені .gov.ua (або на Вашому сайті), зображено на рисунку 4.2.

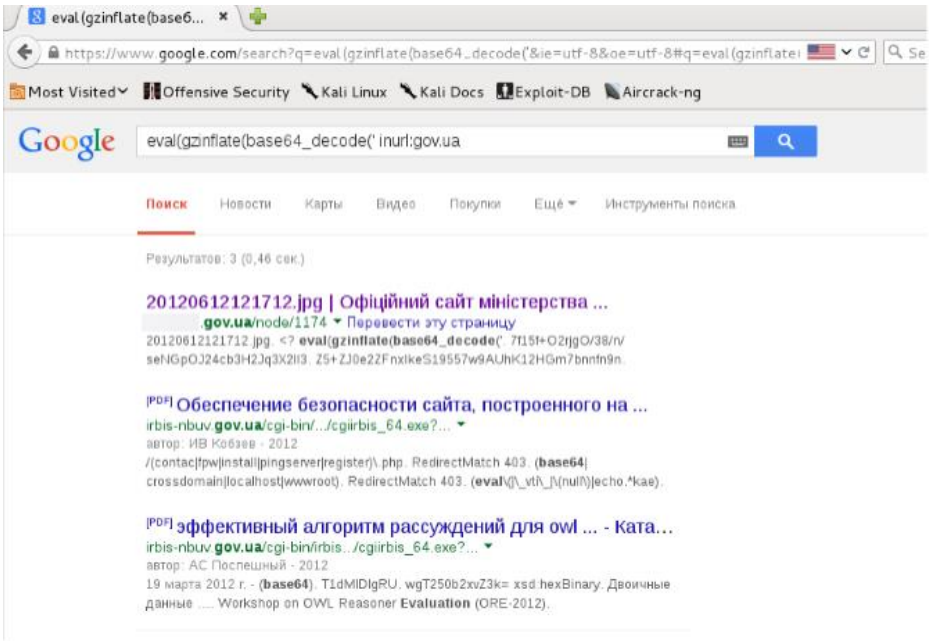

Рисунок 4.2 – Пошук шел у домені .gov.ua

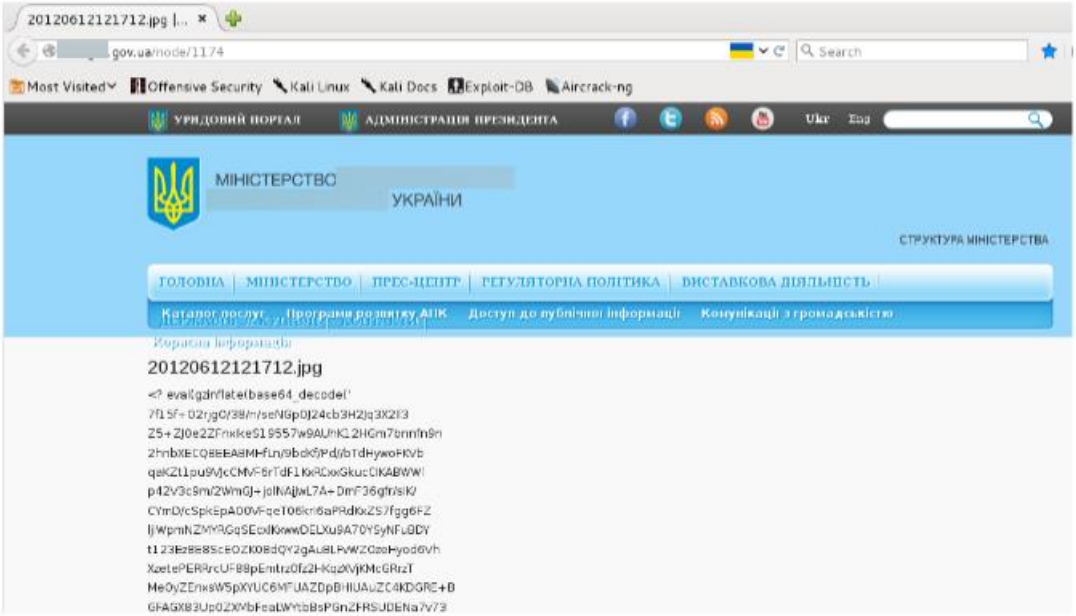

Рисунок 4.3 – Пошук шел у домені .gov.ua

Копіюємо його у текстовий файл та зберігаємо для аналізу та подальшої передачі до CERT-UA за допомогою Форми повідомлення про інцидент. Аналіз можливо провести і самостійно, наприклад використавши онлайн-сервіси з декодування та деобфускації (важливо розуміти, що таким чином Ви дасте знати Інтернету про те, що вам відомо про шел). Приклад такої онлайн-деобфускації – тут, а вихідний код шелу можна знайти тут. Отримання тіла шелу, його ідентифікація (наприклад, за md5-сумою) та вивчення може дати інформацію про вразливість, яка біла експлуатована зловмисником. Крім того, можна дізнатись про шлях протидії повторній експлуатації цієї ж вразливості. Часто адміністратори, намагаючись швидше приховати факт зламу, не вживають необхідних заходів з закриття вразливості, що стала причиною зламу (просто, відновивши весь сайт з резервної копії) і злам повторюється.

Індивідуальні завдання

1. Реалізувати пошук веб-шелів на сайті, який описано в практичній роботі.

2. Обрати сайт однієї із шкіл міста Луцька, згідно варіанту та здійснити пошук веб-шелів на сайті:

– наявність реклами, спливаючих вікон;

– сканування сайту на предмет ознак компрометації;

– перевірка та пошук підозрілих файлів сайту.

3. Реалізувати пошук веб-шелів на сайті, яким користуєтеся особисто Ви найбільше.

4. Порівняти результати та зробити висновки.

ЗАВДАННЯ, ЯКЕ ОЦІНЮЄТЬСЯ МАКСИМАЛЬНО:

1. Підготовка середовища

Розгорніть тестовий веб-сервер (Apache/Nginx + PHP) у віртуальній машині або контейнері.

Створіть «заражений» сайт, додавши кілька типових веб-шелів (наприклад, простий eval-скрипт із шкідливими командами).

2. Виявлення шелів

Налаштуйте антивірус або спеціалізовані сканери (ClamAV, LMD, WAF-модулі).

Виконайте пошук підозрілих файлів за критеріями:

– розширення (.php, .phtml, .asp),

– вміст (рядки eval, base64_decode, exec),

– права доступу та дати зміни.

Занотуйте всі знахідки.

3. Аналіз логів

Перегляньте access- та error-логи веб-сервера.

Визначте джерело завантаження шелу (IP, користувач, час).

4. Ліквідація

Видаліть або ізолюйте шкідливі файли.

Перевірте цілісність основних директорій сайту.

Оновіть CMS / плагіни, змініть облікові дані.

5. Профілактика

Увімкніть WAF або модулі типу mod_security.

Налаштуйте обмеження прав для веб-користувача.

Складіть короткий чек-лист «Що робити після інциденту».

Зміст звіту

1. Тема та мета практичного заняття.

2. Результати у виді скріншотів або таблиць щодо одержаних результатів завдань.
3. Висновки.
4. Відповіді на контрольні запитання.

Контрольні запитання

1. Що таке веб-шел?
2. Які ознаки можуть вказувати на наявність веб-шелу на сервері?
3. Назвіть базові кроки з пошуку веб-шелів.
4. Хто такий «інсайдер» у контексті кібербезпеки?
5. Назвіть три основні мотиви інсайдерських дій.
6. Як організація може швидко виявити підозрілу поведінку співробітника?
7. Що робити після інциденту, пов’язаного з інсайдером?

Література: [2, 3, 10, 11, 12, 14]

Практична робота №5.
Одержання інформації облікових записів в Facebook. Використання можливостей фреймворку Gophish для фішингових атак

Мета: виявити можливості методик по розкриттю електронних адрес та даних профілів у соціальній мережі. Виявити потенційні небезпеки, які із цим пов'язані. Зробити висновки про превентивні заходи по уникненню відповідних можливостей

Короткі теоретичні відомості

Основна ідея OSINT – цілеспрямований збір інформації (harvesting) щодо об'єкта зацікавленості з метою подальшої обробки та різновекторного контент-аналізу отриманих даних та інтерпритації кінцевих результатів (створення інформаційного «портрету» особи, виявлення неочевидних фактів чи зв'язків, прогноз її ймовірної поведінки тощо).

При цьому, зібрана інформація має обов'язково проходити процес фільтрації шляхом оцінки надійності джерел її походження й достовірності самих відомостей.

Основою для пошуку може стати профіль особи в соціальній мережі, його прізвище та ім'я, сфера громадської активності або назва компанії, певна фотографія або відеоконтент, веб-сайт чи адреса електронної пошти.

Алгоритм пошукової діяльності:

1. Визначення вихідних даних та формулювання мети пошуку - що відомо про об'єкт (прізвище та ім'я, вік, телефон, фото, посада тощо) та що необхідно встановити, якої інформації не вистачає (приміром, де особа перебувала в певний момент часу, хто є реальним власником компанії та ін.);

2. Визначення необхідних інструментів та методів OSINT універсальний набір інструментів і алгоритм дій визначити складно – ключові пошукові слова, джерела потенційної інформації та задіяний інструментарій залежать від вихідних даних та кінцевої мети пошуку;

3. Здійснення пошуку інформації та аналіз зібраних відомостей пошук може бути пасивний (збір інформації з пошукових систем, у тому числі за фото- та відео; аналіз персональних даних та активності користувача в соціальних мережах і блогах, на форумах чи в месенджерах; отримання геолокаційних даних за допомогою

загальнодоступних ресурсів) та активний (збір даних на закритих ресурсах, доступ до яких можливий лише за передплатою; застосування спеціалізованих сервісів та програм).

Методики, які використати у цій роботі:

Методика 1:

1. Задатись потенційною адресою користувача в домені gmail.

2. У власній gmail-скриньці апробувати валідність адреси шляхом задання її у драфті листа.

3. Скопіювати 3 фото з аватарки користувача, який існує в Google.

4. Скориставшись сервісом розпізнавання віку за фото (можна використати https://age.toolpie.com/ або інший ресурс) визначити потенційний вік цільової особи.

Знання вікової категорії дає можливість соціальному інженеру використовувати в своїх цілях особливості та інтереси, притаманні типовому представнику відповідного покоління, а також маніпулювати знаннями про вік при складанні таргетованих листів.

Зв’язок із профілем користувача у Facebook (в тому числі із прихованими параметрами) також може дати корисну інформацію для соціального інженера.

Методика 2:

1. Одержати ID таргетованого профілю. Для цього перейти на сторінку профілю, обрати “перегляд вихідного коду” (або inspect element) та виконати пошук «UserID» (без лапок). Результат виглядатиме як «UserID»:10000xxxxxxx.

2. Виконати пошук 3 спільних друзів між двома профілями (один із яких може бути Ваш), де номери 1 та 2 представляють собою номери ID відповідних профілів.

3. Пересвідчитись, чи працює даний спосіб коли список “друзів” в одному із профілів встановлений як приватний.

4. Пересвідчитись, чи працює даний спосіб для виявлення спільної множини друзів довільних профілів.

Методика 3:

1. Створити нову сторінку (рис.5.1), адмініструвати (рис.5.2) та налаштувати (рис.6.3) її на основі вашого профілю.

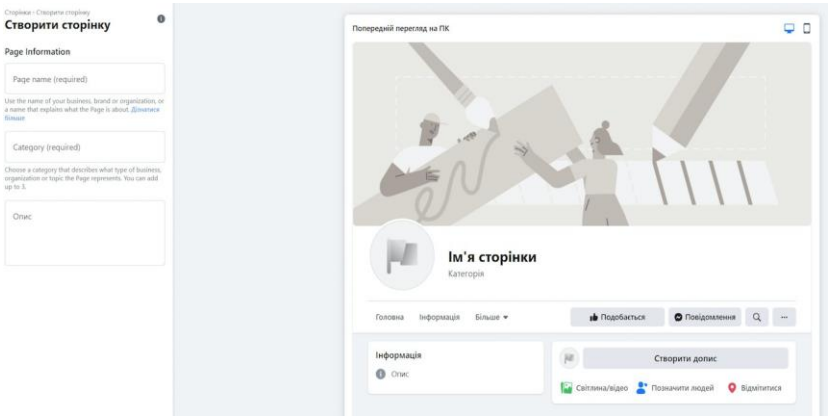

Рисунок 5.1 – Створення нової сторінки

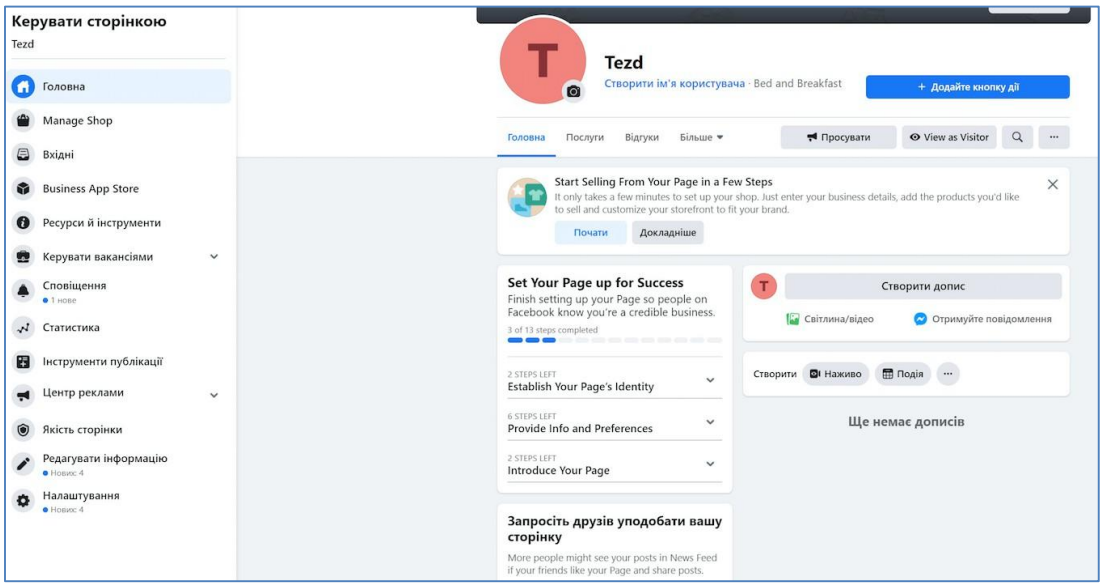

Рисунок 5.2 – Адміністрування сторінки

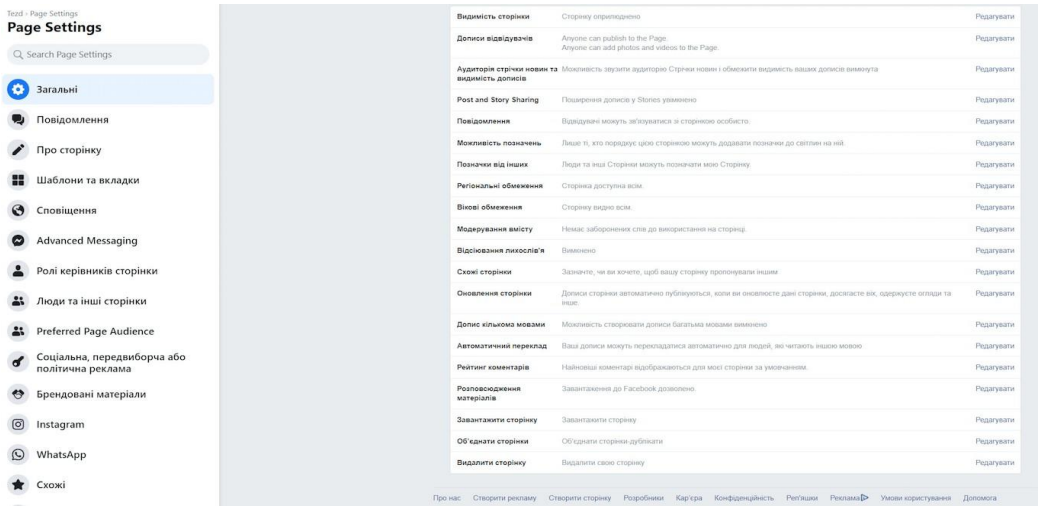

Рисунок 5.3 – Налаштування

2. Спробувати додати когось як адміністратора цієї сторінки (рис.5.4.)

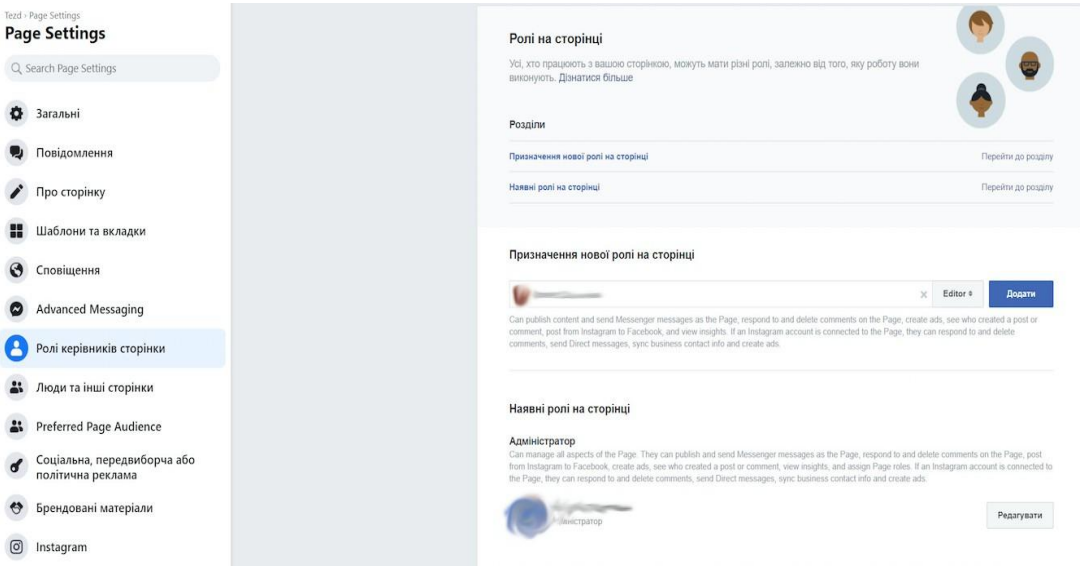

Рисунок 5.4 – Надання ролі

3. Відкрити інструменти для розробки та використати аналітику Facebook по одержанню даних зі зв’язаного акаунту (рис.5.5.).

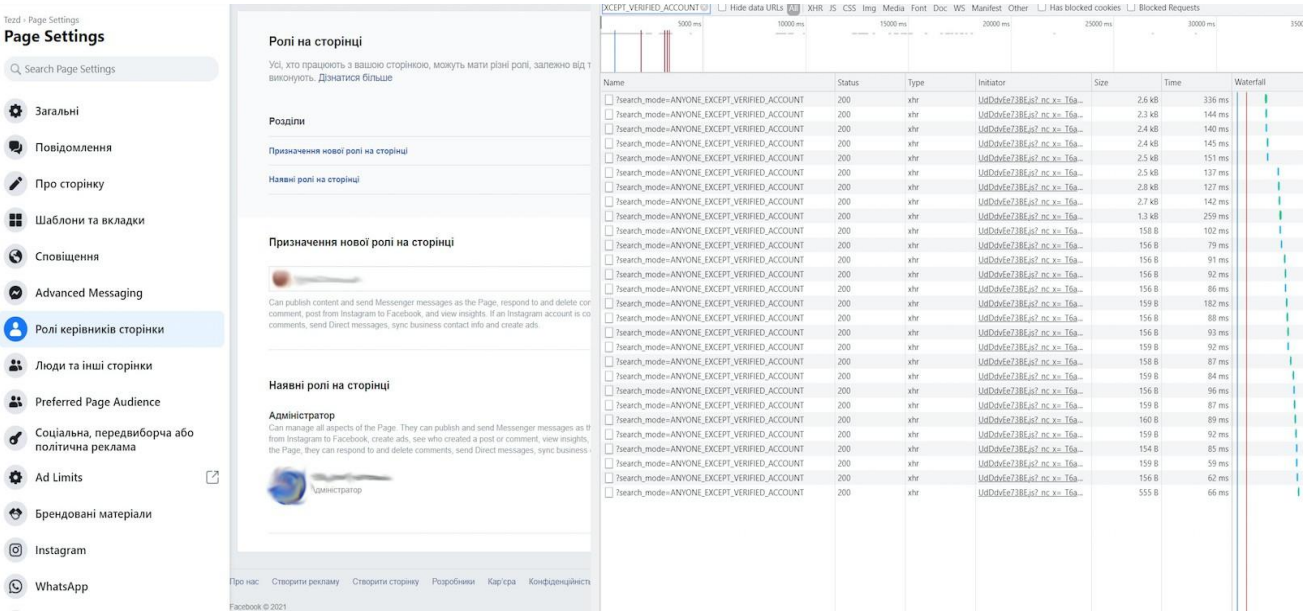

Рисунок 5.5 –Аналітика подій зв’язаного акаунту

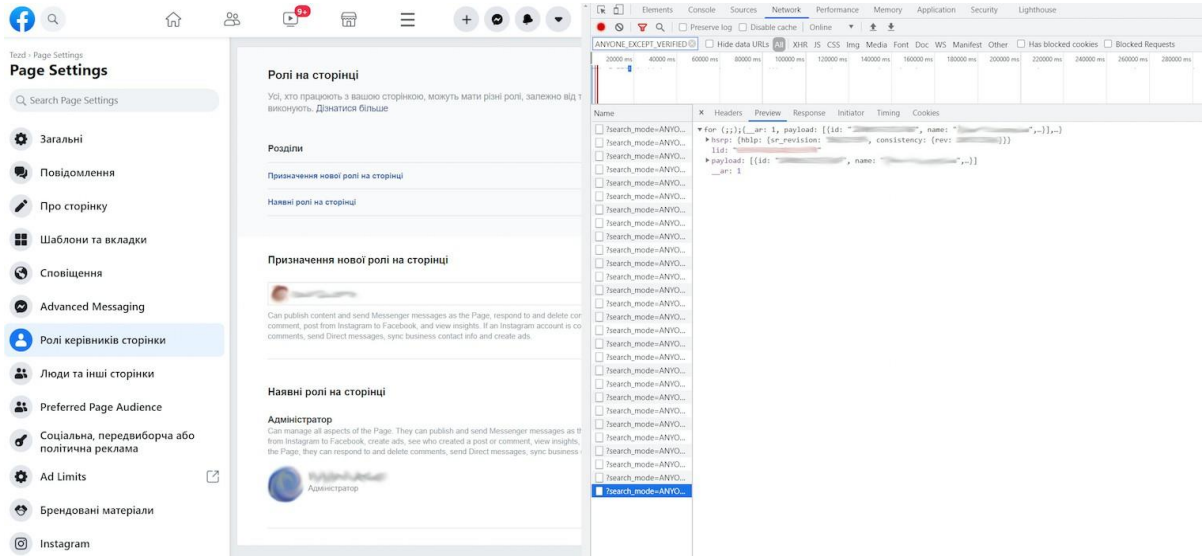

Рисунок 5.6 – Одержання даних про ID та інші параметри

Методика 3, яка описана в прктичній роботі пов'язана із використанням автоматизованого засобу для здійснення фішингових атак Gophish. Однак, треба усвідомлювати відповідальність за надсилання нав'язливих чи фішингових листів дописувачам. Тому для тестування можливостей фреймворка слід задатись множиною адрес користувачів, які згодні брати участь в експерименті.

Gophish – це фішинговий фреймворк, який надзвичайно спрощує симуляцію реальних фішингових атак. Ідея gophish проста – зробити навчання фішингу галузевого рівня доступним для всіх. «Доступний» у цьому випадку означає дві речі:

Доступна ціна – Gophish – це програмне забезпечення з відкритим кодом, яке є повністю безкоштовним для використання будь-ким.

Доступність – Gophish написаний мовою програмування Go. Перевага полягає в тому, що релізи gophish є скомпільованими бінарними файлами без залежностей. Коротко кажучи, це робить встановлення таким же простим, як «завантажити та запустити»!

Детальніша інформація про **Gophish** - https://docs.getgophish.com/user-guide/

Індивідуальні завдання

1. Задатись необхідними вихідними даними: нікнеймом, іменем користувача який є метою дослідження «соціального інженера» та іншими.

2. Проробити методики 1 та 2 згідно наведених теоретичних відомостей з трьома

користувачами зі скріншотами відповідних етапів.

3. Розібратися з роботою методики 3 теоретично та описати як створити нову сторінку та налаштувати її на основі вашого профілю. Чи є негативні наслідки цього методу.

4. *При можливості завантажити Gophish. Створити компанію із 3 одногрупників, які дали дозвіл на отримання фішингових листів.

Описати як виконувалися наступні дії:

– зміна налаштувань облікового запису;

– групи;

– шаблони;

– відстеження вкладень;

– цільові сторінки;

– надсилання профілів;

– кампанії;

– використання API;

– генерація звітів;

– звітність електронною поштою;

– вебхуки;

– керування користувачами.

5. Зробити висновки про розкриття даних, та запоруки успішності відповідних трьох методик.

Зміст звіту

1. Тема та мета практичного заняття.

2. Результати у виді скріншотів та поясненнями щодо одержаних результатів завдань.

3. Надати рекомендації по запобіганню діям соціального інженера.

4. Відповіді на контрольні запитання.

Контрольні запитання

1. Які превентивні дії по прив’язці акаунтів соціальних мереж до поштових адрес ви можете порекомендувати?

2. Що таке «патерн» електронної пошти організації? Яким чином він може бути використаний зловмисником?

3. Які шляхи опосередкованого одержання приватної адреси особистості може використовувати соціальний інженер?

4. В чому є небезпека creeper`ів в соціальних мережах?

5. Які дані профілю можуть бути використані соціальним інженером?

Література: [2, 3, 10, 11, 12, 14]

Практична робота №6.
Пошук інформації по цільовому об’єкту засобами відкритої розвідки

Мета: опанувати методику пошуку та інтерпретації інформації, яка цікавить соціальних інженерів, на основі підходів OSINT. Зробити висновки про превентивні заходи

Короткі теоретичні відомості

OSINT (Open Source Intelligence) відрізняється від OSIF (Open Source InFormation). Друге – це дані та відомості, які циркулюють у вільно доступних медіа-каналах. В результаті OSINT ми одержуємо специфічно зібрану та структуровану для відповіді на конкретні питання інформацію. Збір даних в OSINT принципово відрізняється від інших напрямів розвідки, зокрема агентурної розвідки. Головна задача агентурної розвідки – це одержання інформації із джерел, які переважно не націлені на співробітництво. Головна задача OSINT – це пошук змістовних та надійних джерел серед різноманіття загальнодоступної інформації.

Основні етапи еволюції OSINT:

1945-2005 (аналіз відкритих радіо-сигналів та публічних матеріалів, збір іноземних новин тощо);

2005-2009 (CIA (Central Intelligence Agency) запускає Open Source Centre. Початок активізації ролі соціальних інтернет-медіа);

2009-2016 (персональні дані передаються через смартфони та IoT пристрої, і є основними джерелами OSINT)

2016-2019 (OSINT широко використовується в бізнесі, політичних кампаніях, дослідництві, інвестуванні та задачах кібербезпеки).

Основні етапи розвідки:

– створення плану дослідження;

– підготовка програм та обладнання, необхідних для поставленої задачі.

– Виконання пошуку по доступних ідентифікаторах.

– Збір інформації.

– Аналіз одержаних даних.

– Підготовка заключення та результатів.

– Архівування або очистка обладнання.

Приклади пошуку відкритих даних по фото

Задача 1: Знайти, де був зроблений знімок будівлі з дерев'яним фасадом. Поруч припарковані машини, які зображено на рисунках 6.1- 6.3.

Рисунок 6.1 – Пошук Google дає надто велику кількість схожих зображень

Рисунок 6.2 – Результати пошуку схожих зображень

Треба проаналізувати зображення на предмет підказок.

Такими підказками можуть бути номери машин. Вони відповідають Німеччині, регіон Берліна.

Можна напряму вказати шукані ознаки в Google (дерев'яні фасади Берлін). Пошук дав фото із назвами района. Конкретизуємо пошук:

berlin weddig holzwohnung . Після чого було одержане зображення, і точна адреса будинку.

Задача 2. Відновити, який надпис на будівлі знято на фрагменті відео (рис.7.3). Машина проїжджає під віадуком десь у Стамбулі, на якому створено напис «*cumhuriyeti ve ...»* Перекладач Google дає переклад «республіка та...». Яке завершення цього напису?

Рисунок 6.3 – Результати пошуку схожих зображень

1. Зворотний пошук в Google не дав результатів.

2. Можна описати те, що видно на зображенні. Використовуючи лапки “” для точного виконання запиту, щоби обмежити кількість результатів можна сформувати запит так: “istanbul viaduct historical.” Результат запиту дасть множину акведуків, які є у Стамбулі.

3. Щоби відфільтрувати акведуки, можна переформулювати запит: Istanbul viaduct historical –aqueduct (рис.6.4.).

4. Серед множини зображень є схоже, однак надпису на ньому вже нема.

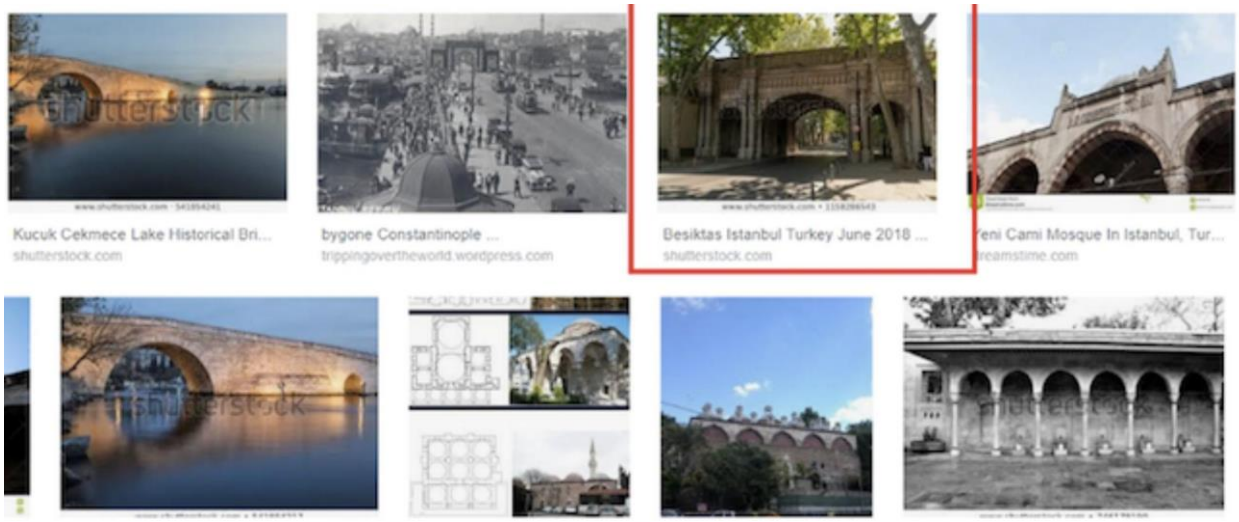

Рисунок 6.4. – Результат запиту пошуку історичного віадуку

Тепер, використовуючи зворотний пошук по зображенню, можна дізнатись про об’єкт, і про надпис, який там був раніше:

“*Cumhuriyeti ve demokrasiyi seviyoruz.*” “Ми за демократію та республіку”.

Також існують інші можливості пошуку за іншими прямими та непрямими ознаками інформації про цільовий об’єкт.

Засоби OSINT для аккаунтів Facebook та пошуку загальної інформації:

graph.tips; whopostedwhat.com; lookup-id.com; @usersbox_bot; @GetPhone_bot; https://lookup-id.com; http://barometer.agorapulse.com; https://www.facebook.com/safety/groups/law/guidelines; https://osint.link/osint-part2/#facebook

Засоби OSINT для аккаунтів Instagram та пошуку загальної інформації:

gramfly.com; codeofaninja.com; sometag.org; archive.org; @InstaBot; @usersbox_bot; undelete.news.

Пошуковики OSINT, які включають модулі для збору інформації про пошту, телефонні номери, домени, ip-адреси, фотографії, людей, рекламні модулі та ін.: https://leak.sx; http://scylla.sh; https://intelx.io; https://4iq.com; https://leaked.site; https://hashes.org; https://leakcheck.io; https://vigilante.pw; https://leakcheck.net; https://weleakinfo.to; https://leakcorp.com; https://leakpeek.com; https://rslookup.com; https://snusbase.com https://ghostproject.f; https://leak-lookup.com; https://nuclearleaks.com; https://private- base.info; https://haveibeensold.app;

https://breachchecker.com; https://www.dehashed.com; http://scatteredsecrets.com; https://haveibeenpwned.com; https://services.normshield.com.

Крім цих засобів є також засоби пошуку по соціальних медіа платформах: https://namechk.com; https://www.namecheckr.com/; https://www.usersearch.org; https://sherlock-project.github.io .

Засоби психологічного аналізу:

Можна використовувати подібні засоби, наприклад, https://sapling.ai/utilities/tone для автоматизованого аналізу текстів, залишених користувачами. Однак, для прицільного аналізу ручна обробка має свої переваги.

Індивідуальні завдання:

1. Існує задача розшукати інформацію про цільову особу та місце її проживання, вподобання тощо. Є фото особи, ПІБ, e- mail.

2. Запропонуйте стратегію пошуку даних по цій особі. В якості цільової особи можна взяти одного з одногрупників/співробітників, які дали на це згоду. Бажано, щоби особа не була з вами у товариських відносинах, для ускладення задачі.

3. Підберіть засоби, які можуть бути ефективно використані вами для задач пошуку.

4. Запропонуйте алгоритм пошуку даних.

5. Виконайте пошук.

6. Виконайте профілювання стилю спілкування особи по методиці DISC (використовуючи пости, дані фото тощо). Обґрунтуйте свої міркування.

7. Зробіть висновки про а)стиль спілкування, який треба використати для створення довіри; б)сферу, з якої може успішніше діяти соціальний інженер, на етапі претекстінгу.

Зміст звіту

1. Тема та мета практичного заняття.
2. Результати у виді скріншотів та поясненнями щодо одержаних результатів завдань.
3. Рекомендації, які можуть запобігти витоку чутливих даних у відкритий доступ.
4. Відповіді на контрольні запитання

Контрольні запитання

1. Що таке претекстінг, які дані для нього потрібні?
2. Які тактики пошуку інформації може використовувати соціальний інженер?
3. Які розділи бажано включити в тренінг по підвищенню обізнаності щодо розміщення інформації у відкритому доступі?
4. Які засоби аналізу доступні аналітику, або ж соціальному інженеру, щодо інформації у відкритому доступі?

Література: [2, 3, 10, 11, 12, 14]

Практична робота №7.
Підбір та підміна облікових даних

Мета: пояснити, що таке підбір облікових даних (brute force / credential stuffing) та які його основні види. Ознайомитися з основними видами атак на облікові дані, навчитися визначати слабкі паролі та способи їх підбору, а також сформувати рекомендації щодо захисту

Короткі теоретичні відомості

Підбір – це спроби вгадати правильні логін/пароль шляхом багаторазових переборів. Є кілька підвидів:

– Brute-force (повний перебір) – послідовне випробовування великого простору паролів (рідко застосовується прямо через захисти, але може бути частиною атаки на слабкі паролі).

– Password spraying – атака, коли атакуючий пробує невелику кількість поширених паролів (напр., 123456, Password1) на велику кількість логінів, щоб уникнути блокувань.

– Targeted guessing – користується персональною інформацією (дати народження, імена домашніх тварин).

Мета: знайти правильний пароль/комбінацію для входу.

Індикатори: багато невдалих логінів з однієї IP або в межах одного аккаунта; зростання помилок авторизації; повторні блокування.

Захист: обмеження спроб входу, lockout, rate-limiting, CAPTCHA, 2FA, обмеження за IP/геолокацією, моніторинг аномалій.

Термін «підміна» використовують у двох близьких значеннях, які часто плутають.

1. Credential stuffing (авторизація через повторне використання злитих паролів)

Атакувальники використовують бази злитих логінів+паролів (breach leaks) і автоматично «підставляють» ці комбінації на інших сайтах/сервісах, де жертва могла повторно використати пароль.

Мета: зайти, використовуючи валідні (але викрадені) облікові дані.

Відмінність від підбору: тут не вгадують — використовують вже відомі пари. Часто дуже швидко й масштабно (багато обліків перевіряються одночасно).

2. Підміна (заміна) облікових даних у акаунті жертви

Це коли нападник вже отримав доступ і змінює email/телефон/пароль (щоб «відрізати» власника від доступу).

Мета: утримати контроль, ускладнити відновлення доступу власником, підготувати подальший шахрайський сценарій.

Індикатори: власник не може зайти, приходять повідомлення про зміну паролю/пошти/реквізитів, незвичні «вихід з усіх сесій», змінені recovery-опції.

Як розпізнати та запобігти атакам підміни облікових даних:

– Якщо бачиш багато неправильних спроб входу → підозра на підбір. Блокуй IP, включай CAPTCHA, застосуй rate limiting.

– Якщо сервер фіксує багато успішних входів із різних IP, особливо з незвичних локацій, та відразу зміну рековері → підміна/credential stuffing або повний takeover. Терміново: скинути паролі, відключити сесії, повідомити користувачів, включити 2FA.

– Якщо отримав повідомлення про зміну email/телефону — це критичний IOC: відновлюй через офіційні канали, звертайся в службу підтримки, змінюй паролі і перевіряй логи.

Захист повинен починатися з паролів. Встановлюйте надійні та унікальні паролі для кожного сервісу. Рекомендовано створювати комбінації щонайменше з 16 символів, що містять великі й малі літери, цифри та символи. Для цього зручно користуватися генераторами паролів, а для зберігання – менеджерами паролів, де ви зберігаєте всі дані під одним головним паролем.

Не менш важливо — ввімкнути MFA для всіх сервісів, де це можливо. Навіть якщо зловмисник дізнається ваш пароль, без другого фактора він не зможе отримати доступ до акаунта

Індивідуальні завдання:

1. Навести приклади слабких паролів і пояснити, чому вони небезпечні.

2. Скласти приклад словника паролів для тестування.

3. Пояснити, як можна захиститися від підбору паролів.

4. Скласти складний пароль, використовуючи генератор паролей (наприклад, сервіс « HaveIBeenPwned »).

5. Написати програму перевірки web-сайту на складність паролю, використовуючи PHP та JavaScript написати скрипти обробки сторінки регістрації, на якій присутні:

– Прізвище, ім’я та по-батькові (українською),

– поштова адреса,

– адреса електронної пошти,

– телефон (стаціонарний),

– телефон (мобільний) з кодом країни.

Зміст звіту

1. Тема та мета практичного заняття.

2. Результати у виді скріншотів та поясненнями щодо одержаних результатів завдань.

3. Рекомендації, які можуть запобігти підбору та підміні облікових даних.

4. Відповіді на контрольні запитання

Контрольні запитання

5. Що таке підбір облікових даних ?

6. Які види підбору Вам відомо?

7. Що таке підміна (заміна) облікових даних у акаунті жертви?

8. Як розпізнати та реагувати на підбір та підміну облікових даних?

9. Пояснити, як можна захиститися від підбору паролів.

Література: [2, 3, 10, 11, 12, 14]

Практична робота №8.
Бекдори у Windows та Android: принципи роботи та методи виявлення

Мета: ознайомитися з поняттям бекдору та його використанням у шкідливому ПЗ. Навчитися виявляти ознаки роботи бекдору у Windows та Android. Розробити заходи захисту від подібних атак

Короткі теоретичні відомості

Що таке бекдор?

1. Прихований доступ: Це спеціально створений або випадково залишений "чорний хід" для несанкціонованого доступу до системи. Бекдор - це тип шкідливого програмного забезпечення або функціональність, яка забезпечує несанкціонований доступ до комп'ютера, мережі або програмного забезпечення.

2. Віддалене керування: Бекдор дозволяє зловмиснику дистанційно керувати пристроєм, як якщо б він був його законним власником. Він дозволяє зловмисникам обходити стандартні процедури аутентифікації та безпеки, отримуючи доступ до системи, навіть якщо інші захисні механізми працюють

3. Непомітність: Головна небезпека бекдору в тому, що він працює непомітно для звичайного користувача, залишаючись прихованим від антивірусних програм та системних перевірок. Бекдори можуть бути впроваджені у легітимне програмне забезпечення або існувати як окреме шкідливе програмне забезпечення.

Основні типи бекдорів:

– Програмні бекдори – встановлюються через шкідливе програмне забезпечення.

– Апаратні бекдори – впроваджуються на рівні апаратного забезпечення.

– Мережеві бекдори – відкривають порти або використовують протоколи доступу до мережі.

– Бекдори ПЗ – створюються зловмисно для доступу.

Бекдор на Android: бекдор, про який вам слід знати, і як захистити себе:

– Бекдор забезпечує прихований, постійний доступ до пристрою та координує роботу із серверами C2.

– Він поширюється через модифіковані програми, маніпульовані пристрої та трояни RAT з модулями на вимогу.

– Це дозволяє використовувати програми-вимагачі, крадіжку даних, ботнети та криптоджекінг без відома користувача.

– Захист: програми з перевірених джерел, патчі, двофакторна аутентифікація (2FA), захист від шкідливого програмного забезпечення та придбання сертифікованого обладнання.

Вразливості iPhone проти шкідливого програмного забезпечення Android: що дійсно важливо для вашої безпеки

– Зловмисники віддають перевагу Android через його масштабованість та відкритість, тоді як iOS вирізняється своєю більш контрольованою, але не надійною екосистемою.

– Більшість мобільних шкідливих програм є троянами; оновлення, обмеження дозволів та запобігання завантаженню несанкціонованих програм значно знижують ризик.

– У підприємствах MDM, Apple Business Manager та Android Enterprise є ключовими для шифрування, політик та видимості.

– Ефективна безпека поєднує дизайн платформи, швидке встановлення патчів та звички користувачів з автентифікацією та верифікацією додатків.

iOS (контрольована екосистема): App Store з суворим оглядом, шифрування за замовчуванням, пісочниця, підпис коду, та такі елементи, як Безпечний анклав, Face ID та Touch ID. Детальні дозволи та розподіл швидкі та одночасні оновлення на сумісні пристрої. У свою чергу, користувач має менше можливостей для налаштування.

Android (відкритість та різноманітність): платформа з відкритим кодом з широкою екосистемою виробників. Багаторівнева безпека: Захистити Google Play аналізує програми, деталізує дозволи, повне шифрування, біометрія та покращення модернізації компонентів. Android Enterprise (робочі профілі, повністю керований режим) та такі технології, як Samsung Knox передбачає потенційні загрози безпеці, автоматично реагувати на них і розширювати захист за межі мобільних пристроїв на IoT-пристрої та мережеві точки кінця.

В обох системах, рішення користувачів та ІТ є вирішальними: активувати оновлення, огляд дозволів, вибирати пристрої з гарною підтримкою, а уникнення невідомих джерел значно зменшує ризик.

Індивідуальні завдання:

1. Пояснити різницю між:

– легітимними віддаленими адміністративними інструментами (RDP, TeamViewer, MDM для Android)

– шкідливими бекдорами (запускаються приховано, без згоди користувача).

2. Навести приклади реальних атак із використанням бекдорів (наприклад, Back Orifice, Gh0st RAT, Pegasus).

3. Проаналізувати Windows: дослідити активні процеси та відкриті порти та знайти «підозрілі процеси» (при можливості використати віртуальну машину). Перевірити журнали подій (Event Viewer). Виявити спроби підозрілих з'єднань.

4. Для Android – перевірити дозволи (доступ до SMS, мікрофону, геолокації), енергоспоживання, трафік додатків. Визначити, які дозволи є зайвими.

5. Скласти список рекомендації щодо захисту для користувача Windows та Android.

Зміст звіту

1. Тема та мета практичного заняття.

2. Результати у виді скріншотів та поясненнями щодо одержаних результатів завдань.

3. Аналіз підозрілих елементів.

4. Висновки щодо методів виявлення бекдорів.

5. Рекомендації щодо захисту.

6. Відповіді на контрольні запитання.

Контрольні запитання

1. Яка різниця між бекдором і троянцем?

2. Назвіть 3 способи персистентності бекдора у Windows.

3. Як працює «fileless» бекдор? Чому він складніший для виявлення?

4. Які особливості архітектури Android впливають на механізми бекдорів (порівняно з Windows)?

5. Що таке C2 (command-and-control) і яку роль він відіграє у бекдорах?

6. Наведіть приклади легітимних інструментів, які зловмисник може використовувати для підтримки бекдора (living off the land).

7. Що таке «dropper» у контексті бекдорів?

8. Як бекдор може отримати права SYSTEM у Windows? Назвіть мінімум один шлях.

9. Поясніть концепцію обфускації коду і чому її використовують у бекдорах.

Література: [1, 3, 4, 6, 7].

Практична робота №9.
Ін’єкційні атаки: SQL-ін’єкція та Cross-Site Scripting (XSS)

Мета: ознайомитись з основними принципами реалізації та захисту від SQL-ін’єкції та атаки Cross-Site Scripting (XSS)

Короткі теоретичні відомості

У веб-додатках основними загрозами безпеці є ін’єкційні атаки, які використовують помилки валідації даних користувача. Найпоширенішими є:

SQL-ін’єкція (SQL Injection) – впровадження шкідливого SQL-коду;

Cross-Site Scripting (XSS) — впровадження шкідливого скрипту JavaScript у веб-сторінку.

Ці атаки можуть призвести до:

– викрадення облікових даних користувачів;

– несанкціонованого доступу до бази даних;

– модифікації веб-контенту;

– повного компрометування системи.

SQL-ін’єкція (SQL Injection)

Принцип роботи

SQL-ін’єкція – це метод, при якому зловмисник вставляє у форму введення даних (наприклад, у поле “Логін” або “Пошук”) фрагмент SQL-коду. Якщо введення не перевіряється, код потрапляє у запит до бази даних і виконується.

Приклад:

```
SELECT * FROM users WHERE username = 'admin' AND password = '12345';
```

Якщо користувач введе:

```
admin' OR '1'='1
```

Запит стане:

```
SELECT * FROM users WHERE username = 'admin' OR '1'='1' AND password = '12345';
```

Цей запит завжди поверне істину, і система пропустить зловмисника без пароля.

Наслідки атаки:

– викрадення паролів, персональних даних, фінансової інформації;

– зміна або видалення таблиць;

– отримання доступу до всієї бази даних;

– встановлення бекдорів або завантаження шкідливих скриптів.

Методи захисту:

1. Використання параметризованих запитів (prepared statements):

2. cursor.execute("SELECT * FROM users WHERE username = ? AND password = ?", (user, password)).

3. Валідація та фільтрація вхідних даних – дозволяти лише допустимі символи.

4. Використання ORM (Object-Relational Mapping) – зменшує прямий контакт із SQL.

5. Обмеження прав доступу – користувач БД не повинен мати права на DROP, DELETE ALL тощо.

6. Регулярний аудит коду.

Атака Cross-Site Scripting (XSS)

Принцип роботи

XSS-атака відбувається, коли зловмисник вводить JavaScript-код у веб-сторінку, який потім виконується у браузері іншого користувача. Це можливо, якщо сайт не фільтрує HTML або скрипти, введені користувачем.

Приклад:

Користувач вводить у коментар:

<script>alert('Ви зламані!');</script>

Якщо сайт виводить цей коментар без перевірки, код виконається у браузері всіх, хто його побачить.

Види XSS:

1. **Stored XSS (збережена)** – код зберігається на сервері (у базі даних) і виконується кожного разу при відкритті сторінки.

2. **Reflected XSS (відображена)** – код передається через URL або форму.

3. **DOM-based XSS** – скрипт впроваджується у DOM (структуру сторінки) через JavaScript, браузер змінює сторінку без перевірки на основі введених даних.

Наслідки XSS:

– викрадення cookies або токенів авторизації;
– підміна контенту сторінки;
– виконання дій від імені користувача;
– поширення шкідливих посилань.

Методи захисту:

1. Екранування (escaping) усіх даних, що відображаються в HTML: <script> замість <script>.

2. Валідація введення – заборона тегів <script>, onload, onclick тощо.

3. Використання Content Security Policy (CSP) — заборона виконання неавторизованих скриптів.

4. HTTPOnly cookies — недоступність cookies через JavaScript.

5. Регулярне оновлення бібліотек та фреймворків

Таблиця 10.1 – Порівняння SQL-ін’єкції та XSS

Ознака	**SQL Injection**	**XSS**
Мета атаки	Компрометація бази даних	Викрадення даних користувачів
Місце впливу	Серверна частина	Клієнтська частина (браузер)
Основна вразливість	Неконтрольоване виконання SQL	Неконтрольоване виконання JavaScript
Захист	Prepared Statements, фільтрація введення	Екранування, CSP, валідація HTML

Висновок

Захист веб-додатків від SQL-ін’єкцій та XSS-атак потребує комплексного підходу: грамотної обробки введення користувача; безпечних методів взаємодії з базами даних; правильного налаштування браузерних політик безпеки.

Навчання безпечному програмуванню і проведення тестів на вразливості (наприклад, OWASP ZAP, Burp Suite) — ключ до зменшення ризику компрометації веб-додатків.

Індивідуальні завдання:

Написати Web-сторінку вразливу до SQL-ін’єкцій та XSS. Промоделювати різні види атак та реалізувати захист від них.

Зміст звіту

1. Тема та мета практичного заняття.

2. Результати у виді скріншотів та поясненнями щодо одержаних результатів завдань.

3. Аналіз ін’єкційних атак.

4. Рекомендації щодо захисту.

5. Відповіді на контрольні запитання.

Контрольні запитання

1. Як відбувається реалізація SQL-ін’єкції у веб-додатку?

2. Наведіть приклад SQL-запиту, вразливого до ін’єкції.

3. Які інструменти використовують для тестування SQL-ін’єкцій?

4. Що відбувається, коли користувач вводить у поле логіну admin' OR '1'='1?

5. Чому важливо обмежувати права користувача бази даних?

6. Чим відрізняються Stored XSS, Reflected XSS і DOM-based XSS?

7. Які дані може викрасти зловмисник через XSS?

8. Що таке «escaping» і яку роль він відіграє у захисті?

9. Як впливає використання HTTPOnly cookies на безпеку веб-додатку?

10. У чому полягає принципова різниця між SQL-ін’єкцією та XSS?

Література: [1, 3, 4, 6, 7].

Практична робота №10.
Ризики безпеки інтернет-додатків та аналіз сайтів

Мета: ознайомитись з OWASP. Виявити сайти вразливі до перехвату даних

Короткі теоретичні відомості

Open Web Application Security Project (OWASP) – це міжнародно визнана організація, яка зосереджується на підвищенні безпеки програмних додатків. Це некомерційний фонд, який об'єднує глобальну спільноту професіоналів – від розробників до експертів з безпеки – щоб зробити свій внесок у сферу безпеки додатків з відкритим вихідним кодом. OWASP пропонує безліч ресурсів, включаючи інструменти, документацію та форуми для полегшення розробки, придбання та обслуговування безпечних додатків.

За своєю суттю OWASP прагне зміцнити довіру до програмного забезпечення шляхом надання дієвої та неупередженої інформації про безпеку додатків. Це дозволяє організаціям у всьому світі виробляти, купувати та керувати програмами, які є не лише функціональними, але й захищеними від безлічі загроз кібербезпеці, які сьогодні переповнюють цифровий світ.

OWASP функціонує як платформа, керована спільнотою, що спирається на досвід різноманітного пулу міжнародних учасників. Він вирізняється своїм прагненням пропонувати всі свої ресурси безкоштовно та відкрито для всіх, хто зацікавлений у покращенні безпеки додатків. Це середовище співпраці підтримується в Інтернеті за допомогою дискусійних форумів, місцевих відділень та роботи з конкретними проектами, а також офлайн за допомогою конференцій та зустрічей.

Вплив і важливість OWASP

OWASP відіграє ключову роль у формуванні того, як організації вирішують питання безпеки додатків. Ось кілька ключових причин, чому OWASP незамінний у сучасній цифровій екосистемі:

– Освіта та обізнаність: OWASP робить значний внесок у сукупність знань про безпеку додатків. Він інформує розробників, персонал служби безпеки та організації про існуючі та нові загрози безпеці за допомогою всебічної документації та інструкцій.

– Інструменти та ресурси: Фонд розробляє та надає доступ до різноманітних інструментів та ресурсів безпеки додатків з відкритим вихідним кодом. Вони відіграють важливу роль в оцінці, вдосконаленні та управлінні безпекою програмних додатків.

– Топ-10 OWASP: Мабуть, одним із найвідоміших внесків OWASP є список OWASP Top 10. Цей документ оновлюється кожні кілька років, щоб відобразити найбільш критичні ризики для безпеки веб-додатків. Це є відправною точкою для організацій, які прагнуть визначити пріоритетність своїх зусиль щодо безпеки.

Поради щодо профілактики OWASP

Щоб використовувати всі переваги ресурсів OWASP, організаціям і розробникам рекомендується наступне:

– **Постійно взаємодійте з ресурсами WASP:** Скористайтеся обширною документацією, інструкціями та найкращими практиками безпечної розробки додатків, доступних через OWASP.

– **Інвестуйте в освіту та навчання:** Постійно навчайте та навчайте команди розробників і безпеки щодо останніх ризиків безпеки, тенденцій і найкращих практик безпеки. OWASP пропонує кілька матеріалів та можливостей для навчання.

– **Беріть участь у спільноті:** Приналежність до спільноти OWASP надає доступ до величезного обсягу знань, дозволяючи окремим особам та організаціям залишатися на крок попереду в безпеці додатків.

OWASP залишається наріжним каменем у сфері безпеки веб-додатків. Сприяючи створенню відкритого середовища для співпраці, вона створила платформу, на якій знання про захист програмних додатків знаходяться у вільному доступі та постійно розвиваються. У міру того, як загрози кібербезпеці стають все більш складними, роль таких організацій, як OWASP, у просуванні та полегшенні безпеки додатків ще ніколи не була такою критичною. Беручи участь в ініціативі OWASP і роблячи свій внесок у неї, технологічна спільнота може продовжувати створювати безпечніші цифрові простори як для організацій, так і для кінцевих користувачів.

Індивідуальні завдання:

1. Проаналізувати OWASP top Ten Project та зробити звіт з описом методів

перешкоджання інтернет-загрозам.

2. Запусти OWASP ZAP та протестувати веб-додатки для виявлення вразливостей. Додаткова інформація по встановленню і тестуванню - https://www.youtube.com/watch?v=di9gjFV4Ouw

3. Зробити звіт та список URL-вразливостей, опис рівнів знайдених вразливостей і детальну інформацію по кожній.

Зміст звіту

1. Тема та мета практичного заняття.
2. Результати у виді скріншотів та поясненнями щодо одержаних результатів завдань.
3. Відповіді на контрольні запитання.

Контрольні запитання

1. Що таке OWASP і яка мета проєкту OWASP Top Ten?
2. Які три категорії OWASP найчастіше зустрічаються у реальних атаках?
3. Яке значення OWASP Top Ten має для етичних хакерів і пентестерів?
4. Як виявити неправильну перевірку прав у вебдодатку?
5. Які приклади неправильних налаштувань безпеки ви знаєте?

Література: [1, 3, 4, 6, 7].

Практична робота №11.
Безпека власної мережі Wi-Fi

Мета: виявлення слабких місць бездротової мережі Wi-Fi

Короткі теоретичні відомості

Різновиди протоколу WPA (Wi-Fi Protected Access) – стандарту захисту бездротових мереж, який замінив застарілий WEP.

1. **WPA (Wi-Fi Protected Access, 2003)**

WPA було впроваджено на початку 2000-х років для заміни застарілого стандарту WEP, який став відомим через свої вразливості. Він додав надійніше шифрування за допомогою TKIP (Temporal Key Integrity Protocol - протокол цілісності тимчасового ключа), який забезпечує кращий захист, ніж WEP.

Перше покоління WPA, створене як тимчасове рішення після зламу WEP:

Шифрування: TKIP (Temporal Key Integrity Protocol).

Аутентифікація: PSK (Pre-Shared Key) або 802.1X (Enterprise).

Основна перевага: динамічна зміна ключів для кожного пакета.

Недолік: TKIP вразливий до сучасних атак — нині вважається застарілим.

Варіанти WPA:

– WPA-Personal (PSK) — пароль спільного доступу, використовується вдома

– WPA-Enterprise — аутентифікація через RADIUS-сервер (802.1X).

Їх відмінність полягає в використовуваних ключах шифрування. У невеликих приватних мережах застосовують статичний ключ довжиною 8 символів, яким може бути кодове слово, пароль, PSK (Pre-Shared Key), що задається в налаштуваннях точки доступу і однаковий у всіх клієнтів цієї бездротової мережі. Такий ключ легко скомпрометувати.

2. **WPA2 (2004–2006)**

WPA2, запущений у 2004 році, став світовим стандартом безпеки Wi-Fi більш ніж на десять років. Він замінив TKIP на більш надійний протокол шифрування під назвою AES (Advanced Encryption Standard), який використовується досі.

WPA2 широко використовувався в маршрутизаторах, ретрансляторах і підключених пристроях. Але, як і будь-яка технологія, він не був бездоганним. У 2017

році вразливість KRACK виявила слабкі місця в WPA2, які дозволили зловмисникам перехоплювати трафік навіть у зашифрованих мережах.

– Найпоширеніший стандарт безпеки Wi-Fi (понад 10 років):

– Шифрування: AES із протоколом CCMP (Counter Mode with CBC-MAC Protocol).

– Аутентифікація: WPA2-PSK (домашні мережі) або WPA2-Enterprise.

– Основна перевага: стійке шифрування AES, захист від більшості атак на TKIP.

Недоліки: уразливість до атак KRACK (Key Reinstallation Attack, 2017), слабкі паролі PSK можуть бути підібрані.

3. **WPA3 (2018–дотепер)**

WPA3 був представлений у 2018 році, щоб усунути недоліки попередніх стандартів і відповідати сучасним вимогам безпеки, особливо з появою смарт-пристроїв, віддаленої роботи та зростаючою чутливістю даних.

Найновіший стандарт, розроблений Wi-Fi Alliance:

– Шифрування: GCMP-256 (Galois/Counter Mode Protocol).

– Аутентифікація: новий метод SAE (Simultaneous Authentication of Equals) замість PSK.

– Переваги: захист від офлайн-атаки підбору пароля; forward secrecy (захист історичних даних навіть після компрометації ключа); шифрування всього трафіку навіть у відкритих мережах (Enhanced Open).

Варіанти:

– WPA3-Personal (для побутових мереж, з SAE);

– WPA3-Enterprise (256-бітове шифрування, корпоративний рівень).

4. **Перехідні режими (Mixed Mode)**

Для сумісності з пристроями, що не підтримують нові стандарти:

WPA/WPA2 Mixed Mode — дозволяє одночасно клієнтам WPA і WPA2;

WPA2/WPA3 Mixed Mode — підтримує нові пристрої з WPA3 і старі з WPA2. Але: знижує загальний рівень безпеки (через слабшу сторону).

Таблиця 11.1 – Порівняння версії WPA

Версія	Рік	Шифрування	Аутентифікація	Рівень безпеки	Статус

WPA	2003	TKIP	PSK / 802.1X	Середній	Застарілий
WPA2	2004	AES-CCMP	PSK / 802.1X	Високий	Стандарт до 2018
WPA3	2018	AES-GCMP-256	SAE / 802.1X	Дуже високий	Актуальний

Висновок. Отже, WPA3 сьогодні – єдиний рекомендований стандарт. WPA2 усе ще використовується, але варто ввімкнути AES і складний пароль. WPA та TKIP більше не забезпечують належного рівня захисту.

Індивідуальні завдання:

1. Потрібно провести аудит безпеки власної мережі Wi-Fi.

2. Виявити слабкі місця в захисті роутера.

3. Перевірити пароль на стійкість до Brute force.

4. Дізнатися версію операційної системи та за можливості знайти exploit для отримання доступу до вашого роутеру.

Зміст звіту

1. Тема та мета практичного заняття.

2. Результати у виді скріншотів та поясненнями щодо одержаних результатів завдань.

3. Відповіді на контрольні запитання.

Контрольні запитання

1. Що таке атака KRACK і яких протоколів вона стосується?

2. Який компонент відповідає за шифрування даних у WPA?

3. Поясніть, що таке CCMP і як воно підвищує безпеку.

4. Як WPA3 забезпечує forward secrecy?

Література: [1, 3, 4, 6, 7]

Практична робота №12.
Профілактика та пом’якшення наслідків атак соціальної інженерії

Мета: навчитись налаштовувати параметри безпеки браузерів для обмеження впливу людського фактору на інформаційну безпеку

Короткі теоретичні відомості

Браузер – це основний інструмент доступу до Інтернету, через який відбувається обмін даними, авторизація в облікових записах та передача конфіденційної інформації.

Неправильні налаштування браузера або неуважність користувача створюють ризики: фішингових атак; несанкціонованого збереження паролів; зараження шкідливим кодом через розширення чи скрипти; витоку персональних даних.

Людський фактор у безпеці браузера

Основні прояви: натискання на шкідливі посилання без перевірки; завантаження підозрілих розширень; ігнорування попереджень безпеки (HTTPS, сертифікати); використання однакових паролів для кількох сайтів.

Мета налаштувань безпеки — мінімізувати ризики, пов’язані з помилками користувача.

Ключові параметри безпеки у сучасних браузерах

1. Конфіденційність:

– Увімкнення режиму "Не відстежувати" (Do Not Track).

– Вимкнення збереження історії, кешу та автозаповнення.

– Використання режиму інкогніто / приватного вікна.

– Регулярне очищення cookies.

2. Контроль контенту:

– Блокування вікон, що спливають (pop-ups).

– Заборона або дозвіл JavaScript лише для перевірених сайтів.

– Вимкнення автоматичного відтворення медіа.

– Перевірка дозволів сайтів (доступ до камери, мікрофона, геолокації).

3. Керування розширеннями:

– Використовувати мінімум плагінів.

– Завантажувати їх тільки з офіційного магазину (Chrome Web Store, Mozilla Add-ons).

– Перевіряти дозволи кожного розширення.

4. Аутентифікація та збереження даних:

– Вимкнути автоматичне збереження паролів у браузері.

– Використовувати зовнішній менеджер паролів (Bitwarden, KeePass).

– Увімкнути двохфакторну автентифікацію для основних акаунтів.

5. Захист з’єднання

– Переконуватись у наявності HTTPS.

– Увімкнути функцію "Завжди використовувати безпечне з’єднання" (Chrome, Edge, Firefox).

– Використовувати DNS через HTTPS (DoH) для захисту від підміни DNS-запитів.

Індивідуальні завдання:

1. Перевірити параметри безпеки браузера – навчитися оцінювати поточний рівень безпеки браузера.

Відкрити браузер (Chrome / Firefox / Edge).

Перейти у Налаштування → Конфіденційність і безпека.

Виконати: Перевірити, чи увімкнено HTTPS-режим за замовчуванням.

Вимкнути збереження паролів.

Очистити кеш і cookies.

Зробити скріншоти перед і після змін.

2. Перевірити безпечності розширень – навчитися виявляти потенційно небезпечні плагіни.

Відкрити список розширень (chrome://extensions/ або about:addons).

Перевірити, які розширення мають доступ до даних на всіх сайтах.

Вимкнути або видалити непотрібні.

Зробити висновок, які плагіни потенційно небезпечні.

3. Встановити DNS over HTTPS – підвищити безпеку запитів DNS.

У Firefox: Налаштування → Загальні → Налаштування мережі → Увімкнути DNS через HTTPS.

У Chrome: Налаштування → Конфіденційність і безпека → Безпека → Використовувати захищені DNS.

Перевірити результат на https://1.1.1.1/help.

4. Захист від фішингу – перевірити, як браузер реагує на шкідливі сайти.

Відкрити тестову фішингову сторінку (наприклад: https://phishingquiz.withgoogle.com).

Проаналізувати, як браузер попереджає про небезпеку.

Визначити, які параметри впливають на це.

5. Створення політики безпеки користувача – розробити індивідуальну конфігурацію браузера.

Скласти коротку таблицю:

Параметр	Поточний стан	Рекомендований стан	Примітка
HTTPS режим	Вимкнено	Увімкнено	Безпечне з’єднання
Cookies	Зберігаються	Автоматичне очищення	—
Паролі	Зберігаються	Заборонити	Використовувати менеджер паролів

Зміст звіту

1. Тема та мета практичного заняття.
2. Результати у виді скріншотів та поясненнями щодо одержаних результатів завдань.
3. Відповіді на контрольні запитання.

Контрольні запитання

1. Як браузер попереджає користувача про фішингові сайти?
2. Що таке Do Not Track і для чого використовується?
3. Що таке кеш браузера і як він впливає на конфіденційність?
4. Чому важливо регулярно очищати cookies і кеш?
5. Чим небезпечні спливаючі вікна (pop-ups)?

6. Як браузери реалізують ізоляцію вкладок (sandboxing) і чому це важливо?
7. Що таке browser fingerprinting і як можна зменшити ризик ідентифікації?

Література: [1, 3, 4, 6, 7].

Практична робота №13.
Заходи протидії соціальній інженерії

Мета: пошук та ліквідації небезпечних функцій

Короткі теоретичні відомості

Соціальна інженерія – це методика впливу на людину з метою отримання доступу до інформації або системи шляхом обману чи психологічної маніпуляції.

Сучасні атаки часто поєднують людський фактор і технічні механізми виконання коду, зокрема через функцію eval у веб-додатках.

Eval – небезпечна функція. Характерними ознаками є: складність налагодження, повільне виконання програми, вразливість до атак. Рядок коду функції eval () часто є злоякісним, якщо його отримати від сторонніх осіб. Зловмисники передають його, щоб змінити вміст сторінки або викрасти дані. Виявляти такі приховані конструкції необхідність і для локальних користувачів, які не мають багато ресурсів на дорогі рішення по безпеці.

Функція eval (від англ. evaluate) використовується у багатьох мовах програмування (JavaScript, PHP, Python) для динамічного виконання рядків коду.

Приклад:

```
eval("alert('Hello, user!')");
```

Цей код виконує те, що передано у вигляді тексту.

У контексті безпеки — це небезпечно, бо зловмисник може вставити шкідливий код, якщо вхідні дані не перевіряються.

Соціальний інженер може:

– надсилати фішингові листи або повідомлення, що спонукають користувача вставити або виконати шкідливий скрипт;

– маскувати код у вигляді безпечного (наприклад, eval(base64_decode(...)) у PHP);

– створювати підроблені скрипти «для перевірки системи», які користувач запускає самостійно.

– У такому разі людський фактор (довіра, цікавість) стає «тригером» для запуску шкідливого eval.

Заходи протидії:

Організаційні (людський фактор)

Підвищення освітньої грамотності працівників щодо коду, що виконується.

Заборона на самостійне копіювання та вставку невідомих скриптів.

Розробка інструкцій безпечного поводження з кодом та посиланнями.

Використання політик least privilege (мінімально необхідні права користувача).

Перевірка джерел інформації – навіть якщо повідомлення виглядає “внутрішнім”.

Технічні

Уникати використання eval у коді — замінити безпечними альтернативами:

у JavaScript: JSON.parse(), функціональні виклики напряму;

у PHP: обробка через include або call_user_func();

Статичний аналіз коду — пошук підозрілих конструкцій eval.

Моніторинг системних журналів (логів) — пошук незвичних виконань.

Антивірус / IDS / WAF — можуть виявляти виклики eval у веб-запитах.

Сегментація доступу — щоб навіть успішне виконання eval не давало контролю над усією системою.

Висновки:

– Eval – це потужна, але небезпечна функція, здатна виконувати довільний код.

– Соціальна інженерія часто використовує довіру людини, щоб змусити її самостійно виконати шкідливий код.

– Поєднання освітніх, технічних та організаційних заходів значно знижує ризики.

– Аналіз коду, фільтрація введення та усвідомлення користувача — ключ до захисту.

Індивідуальні завдання:

1. Аналіз коду з eval

2. Навчитися знаходити можливі бекдори у веб-додатках.

3. Реалізувати пошук на комп’ютері не файлів .php, які містять код мови PHP. Якщо шкідливий код дописано у багатьох файлах командою eval, то знайти ці файли та видалити.

Зміст звіту

1. Тема та мета практичного заняття.

2. Результати у виді скріншотів та поясненнями щодо одержаних результатів завдань.

3. Відповіді на контрольні запитання.

Контрольні запитання

1. Яке призначення функції eval у мовах програмування?

2. У чому полягає небезпека конструкції eval(base64_decode(...)) у PHP?

3. Що спільного між eval, exec і system функціями?

4. Що може свідчити про наявність бекдору в коді з eval?

5. Які основні кроки потрібно зробити при виявленні підозрілого коду з eval?

Література: [1, 3, 4, 6, 7].

Практична робота №14.
Заходи протидії соціальній інженерії

Мета: вивчити вразливості мережевих проколів та некоректного налаштування DNS-серверів та NTP-серверів. Вразливості, пов'язані з використанням протоколів SNMP, SSDP та NetBIOS

Короткі теоретичні відомості

DNS (Domain Name System) і NTP (Network Time Protocol) – інфраструктурні служби Інтернету. Їхня цілісність, доступність і достовірність критично впливають на роботу мережі, безпеку з'єднань і коректність журналів. Некоректні налаштування або відомі вразливості цих сервісів відкривають шлях до перехоплення трафіку, DDoS-атак, маніпуляцій часом і компрометації аутентифікації.

Некоректне налаштування DNS-серверів

Сервери DNS (система доменних імен) перетворюють доменні імена (наприклад, example.com) в IP-адреси. Неправильне налаштування може відкрити шлях для серйозних атак.

DNS-спуфінг та отруєння кешу: зловмисник може впровадити в кеш DNS-сервера фальшиві записи, перенаправляючи користувачів на підроблені сайти. Це може призвести до крадіжки облікових даних, фішингу або поширення шкідливого програмного забезпечення.

DDoS-атаки з підсиленням (DNS-amplification): атакуючий відправляє DNS-запит на сервер, використовуючи підроблену IP-адресу жертви. Сервер відповідає жертві, надсилаючи значно більший обсяг даних, ніж був у запиті, що може призвести до перевантаження її мережі.

Зональні перенесення (Zone transfers): якщо не обмежити доступ, зловмисники можуть запросити повний список записів доменної зони, що розкриває внутрішню структуру мережі.

Незахищеність від динамічних оновлень: якщо DNS-сервер дозволяє анонімні динамічні оновлення, зловмисник може додати або змінити DNS-записи, перенаправляючи домен на власні сервери.

Поширені вектори атак

Cache poisoning (отруєння кешу) – підміна cached-відповідей резолвера, через що користувачі потрапляють на фальшиві ресурси.

DNS spoofing / hijacking – переведення домену на інші IP (на рівні реєстратора, реєстр/NS або локального резолвера).

Open resolver abuse – використання відкритих резолверів для DNS amplification DDoS.

Zone transfer (AXFR) misconfiguration – необмежений доступ до повного вмісту DNS-зони (утечка внутрішніх записів).

DNS rebinding – змушування браузера здійснювати запити до внутрішніх ресурсів жертви, обходячи Same-Origin Policy.

Wildcard / NXDOMAIN hijacking – підміна поведінки для неверифікованих доменів (редірект на рекламні/шкідливі сторінки).

Відсутність/некоректне застосування DNSSEC – неможливість криптографічно перевірити достовірність відповіді.

Некоректне налаштування NTP-серверів

Сервери NTP (протокол мережевого часу) використовуються для синхронізації часу в мережі. Неправильне налаштування також може створити вразливості.

NTP-спуфінг: зловмисник може підробити NTP-повідомлення, щоб змусити клієнтів синхронізуватися з невірним часом. Це може мати такі наслідки:

Збій систем: некоректний час може порушити роботу систем, які залежать від точного часу, наприклад, систем ведення журналів.

Обхід механізмів безпеки: деякі механізми безпеки, що ґрунтуються на тимчасових мітках, можуть бути обійдені (наприклад, системи аутентифікації на основі часових ключів).

DDoS-атаки з підсиленням (NTP-amplification): аналогічно DNS, атака може використовувати особливості NTP для підсилення трафіку та перевантаження жертви.

Вразливості програмного забезпечення: застаріле програмне забезпечення NTP-серверів може містити відомі вразливості, які можуть бути використані зловмисниками для отримання доступу або виклику збою.

Поширені вектори атак

NTP amplification DDoS – зловмисник використовує функції NTP (старі команди, наприклад monlist) для посилення трафіку проти жертви.

Time spoofing / spoofed NTP responses – підміна часу, що подає неточний час клієнту/серверу.

Open NTP servers / unauthenticated queries – дозволяють будь-кому отримувати великі відповіді.

Вразливості у старих реалізаціях (буферні переповнення, remote code execution у старих ntpd-версіях).

Як захиститися від вразливостей

Мережеві протоколи:

Використовувати засоби, що запобігають спуфінгу.

Регулярно оновлювати програмне забезпечення мережевого обладнання, щоб усувати виявлені вразливості.

Застосовувати міжмережеві екрани (брандмауери) для фільтрації шкідливого трафіку.

DNS-сервери:

Використовувати DNSSEC (DNS Security Extensions) для перевірки цілісності DNS-записів.

Обмежити доступ до зонних перенесень та динамічних оновлень.

Використовувати фільтрацію трафіку на основі репутації доменів.

NTP-сервери:

Впровадити аутентифікацію NTP для запобігання підміні.

Обмежити доступ до NTP-серверів ззовні, щоб запобігти DDoS-атакам з підсиленням.

Регулярно оновлювати ПЗ NTP-серверів.

Індивідуальні завдання:

1. Використовуючи інтерфейс Windows з метою блокування вразливості, провести відповідне налаштування DNS.

2. Провести перевірку того чи працює на комп’ютері NetBIOS.

Зміст звіту

1. Тема та мета практичного заняття.
2. Результати у виді скріншотів та поясненнями щодо одержаних результатів завдань.
3. Відповіді на контрольні запитання.

Контрольні запитання

1. Чим відрізняється DOS атака від DDOS атаки?
2. В чому типові вразливості NTP-серверу?
3. В чому вразливості протоколу SSDP?
4. В чому типові вразливості DNS-серверу?
5. Які загальні принципи захисту мережевих протоколів?

Література: [1, 3, 4, 6, 7].

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Соціальна інженерія. URL: https://termin.in.ua/sotsialna-inzheneriia/#Falsivij_ antivirus (дата звернення: 08.02.2026).

2. Соціальна інженерія. URL: https://hackyourmom.com/kibervijna/ soczialna-inzheneriya/ (дата звернення: 11.02.2026).

3. Стьопочкін І. В, Ільїн К. І. Теорія та методи соціальної інженерії в кібербезпеці: навч. посіб. для студентів спеціальності 125 «Кібербезпека та захист інформації». Київ: КПІ ім. Ігоря Сікорського, 2023. 35 с. URL: https://ela.kpi.ua/handle/123456789/67176 (дата звернення: 12.02.2026).

4. Зоренко Д. С., Лех Р. В., Кулик Д. О., Червяков О. І. Використання інструментів та методів OSINT для отримання пошукової інформації: практичний порадник. Харків: Інститут підготовки юридичних кадрів для Служби безпеки України, 2023. 36 с URL: https://dspace.nlu.edu.ua/jspui/ bitstream/123456789/19712/1/P_OSINT.pdf (дата звернення: 14.02.2026).

5. Міскевич О. І. Дослідження загроз від кібератак та захист персональної інформації. *Комп'ютерно - інтегровані технології: освіта, наука, виробництво*. 2021. Вип. 45. С. 84-89.

6. Міскевич О. І. Аналіз роботи мережевих утиліт в командному вікні Windows. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2023. Вип. 50. С. 84-89.

7. Що таке Бекдор? Визначення, приклади, бекдор-атаки. URL: https://gridinsoft.ua/backdoor (дата звернення: 22.02.2026).

8. Що таке підбір облікових даних? URL: https://corewin.ua/blog/what-is-credential-stuffing (дата звернення: 22.02.2026).

9. Типи та приклади шпигунського ПЗ. Мобільні шпигунські програми. URL: https://gridinsoft.ua/spyware (дата звернення: 26.02.2026).

10. Онлайн-блог. URL: https://surl.li/blog/uk (дата звернення: 01.03.2026).

11. Онлайн-курс. URL: https://www.udemy.com/ (дата звернення: 01.03.2026).

12. Персональна гігієнга. URL: https://osvita.diia.gov.ua/courses/personal-cyberhygiene (дата звернення: 02.03.2026).

13. Хакінг у практичному застосуванні та соціальна інженерія. URL: https://hackyourmom.com/kibervijna/nastupalna-soczialna-inzheneriya-pidgotovka-do-ataky-chastyna-3/ (дата звернення: 02.03.2026).

14. Кібергігієна: як захиститися від фішингу. URL: https://osvita.diia.gov.ua/courses/kibergigiena-ak-zahistitisa-vid-fisingu (дата звернення: 02.03.2026).

15. Що таке OWASP? ТОП 10 вразливостей. URL: https://training.qatestlab.com/blog/technical-articles/what-is-owasp-top-10-vulnerabilities/ (дата звернення: 02.03.2026).

С59 **Соціальна інженерія**: методичні вказівки до практичних занять для здобувачів першого (бакалаврського) рівня вищої освіти галузь знань 12 (F) Інформаційні технології денної та заочної форм навчання / уклад. О. І. Міскевич. Луцьк: ЛНТУ, 2026. 75 с.

Методичні вказівки до практичних занять з дисципліни «Соціальна інженерія» складені відповідно до діючої програми курсу.

Призначені для здобувачів вищої освіти галузі знань 12 (F) Інформаційні технології.

Комп’ютерний набір О. І. Міскевич

Редактор О. І. Міскевич

Підп. до друку «___» _____ 2026р.
Формат 60х84/16. Папір офс. Гарнітура Таймс.
Ум. друк. арк. ____. Тираж 10 прим. Зам. ____

Відділ іміджу та промоцій
Луцького національного технічного університету
43018, м. Луцьк, вул. Львівська, 75