Дрогобицький державний педагогічний університет
імені Івана Франка

кафедра фізики та інформаційних систем

Роман ЛЕШКО

Кібербезпека

Методичні вказівки для лабораторних робіт, воркшопів, індивідуальних завдань та командних проєктів

Дрогобич
2025

З М І С Т

Вступ

Кібербезпека — це одна з найважливіших складових сучасного цифрового світу. З кожним роком обсяг даних, що обробляється та зберігається в мережах, стрімко зростає, а ризики кібератак стають дедалі складнішими та витонченішими. У світі, де технології стали невід'ємною частиною життя, здатність захищати інформацію від загроз стає не просто навичкою, а необхідністю. Завдання кібербезпеки — не лише реагувати на загрози, але й запобігати їм, будуючи надійні системи захисту. Саме це робить кібербезпеку фундаментальним елементом будь-якої цифрової екосистеми.

Цей підручник створено для того, щоб надати студентам і фахівцям практичні знання та навички в галузі кібербезпеки. Він містить лабораторні роботи, які допоможуть закріпити теоретичні знання, а також плани воркшопів, індивідуальні завдання і командні проекти для розвитку комплексного підходу до вирішення задач. Такий формат навчання дозволяє не тільки зрозуміти основи кібербезпеки, але й застосовувати їх у реальних сценаріях. Адже сучасний фахівець з кібербезпеки — це не лише аналітик загроз, але й стратег, здатний прогнозувати та попереджати ризики. Підручник орієнтований на інтеграцію теорії та практики, щоб забезпечити ефективне навчання.

Ми живемо в епоху, коли безпека інформації торкається кожного аспекту нашого життя: від особистих даних до глобальних мереж. Вивчаючи цей курс, ви станете частиною професійної спільноти, яка стоїть на захисті цифрового простору. Лише завдяки наполегливій праці та прагненню до вдосконалення ми можемо забезпечити стійкість систем перед

обличчям загроз. Тому цей підручник покликаний не тільки навчити вас технічних навичок, але й сформувати критичне мислення та відповідальність за прийняття рішень у сфері кібербезпеки. Досліджуйте, експериментуйте, навчайтеся — і ви станете тими, хто змінює цифровий світ на краще.

Частина 1. Лабораторні роботи

Лабораторна робота № 1. Створення і збереження паролів

Мета: Ознайомитися з основами створення та збереження паролів, вивчити принципи безпечного управління паролями, навчитися створювати програми або алгоритми для генерації та зберігання паролів із використанням сучасних засобів шифрування та захисту даних.

Обладнання: Персональні комп'ютери з встановленою операційною системою (Windows/Linux/macOS).

Програмне забезпечення: програма-менеджер паролів KeePassXC, google-менеджер паролів у Chrome, інтернет ресурси Pwned Passwords .

Теоретичні відомості

Пароль – це набір символів (літери, цифри, спеціальні символи), який використовується для підтвердження автентичності користувача або доступу до захищених ресурсів. Пароль є одним із найпоширеніших (але не єдиним) засобів аутентифікації у сучасних інформаційних системах. Паролі забезпечують захист даних, обмежуючи доступ до конфіденційної інформації, і відіграють важливу роль у контролі доступу, дозволяючи ідентифікувати користувача та надавати йому відповідний рівень доступу. Вони також сприяють підтримці приватності, запобігаючи несанкціонованому доступу до облікових записів і персональних даних.

Паролі відіграють важливу роль у забезпеченні інформаційної безпеки, тому до них висуваються особливі

вимоги. По-перше, вони повинні бути складними, щоб унеможливити їхнє легке відгадування чи підбір зловмисниками. Це означає, що якісний пароль має бути довгим і включати в себе різноманітні символи: великі та малі літери, цифри, а також спеціальні символи. По-друге, важливим аспектом є унікальність паролів. Кожен обліковий запис повинен мати свій окремий пароль, оскільки використання одного і того ж пароля для різних сервісів збільшує ризик компрометації всіх облікових записів у разі витоку одного з них.

Крім того, критично важливим є безпечне зберігання паролів. Паролі ніколи не повинні зберігатися у відкритому вигляді чи в незахищених файлах. Замість цього, рекомендується використовувати надійні менеджери паролів або бази даних із шифруванням. Усі паролі в таких системах мають бути зашифровані сучасними криптографічними алгоритмами, що значно ускладнює їх розшифрування у випадку доступу зловмисників до бази даних. Таким чином, відповідність цим вимогам підвищує загальний рівень безпеки облікових записів і даних.

Однією з програм, що дає змогу реалізувати вище перелічені критерії (і не тільки) це KeePassXC. **KeePassXC** – це відкритий менеджер паролів, який дозволяє зберігати паролі у зашифрованій базі даних. Основні переваги KeePassXC:

а) безпека (використання сучасних криптографічних алгоритмів для шифрування бази даних);

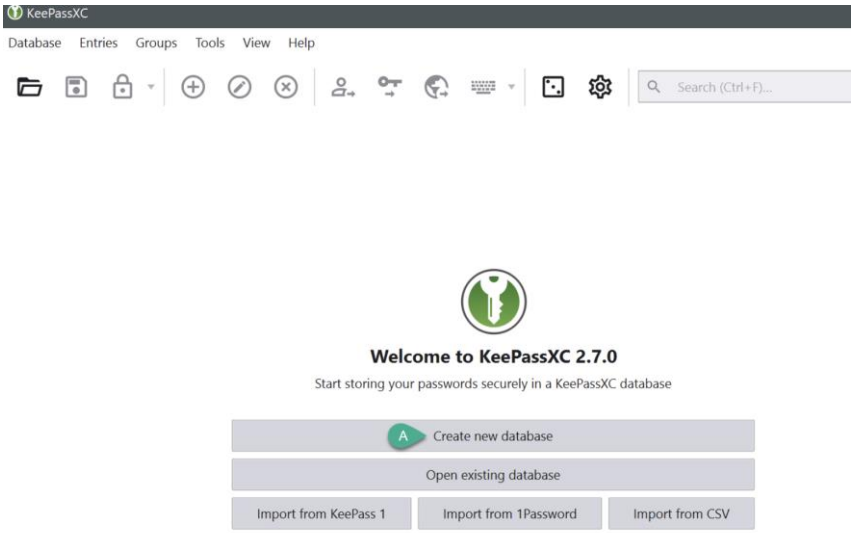
б) універсальність (працює на різних платформах, Windows, macOS, Linux);

в) функціональність (можливість генерації складних паролів, організації їх у категорії та автоматичного заповнення полів входу);

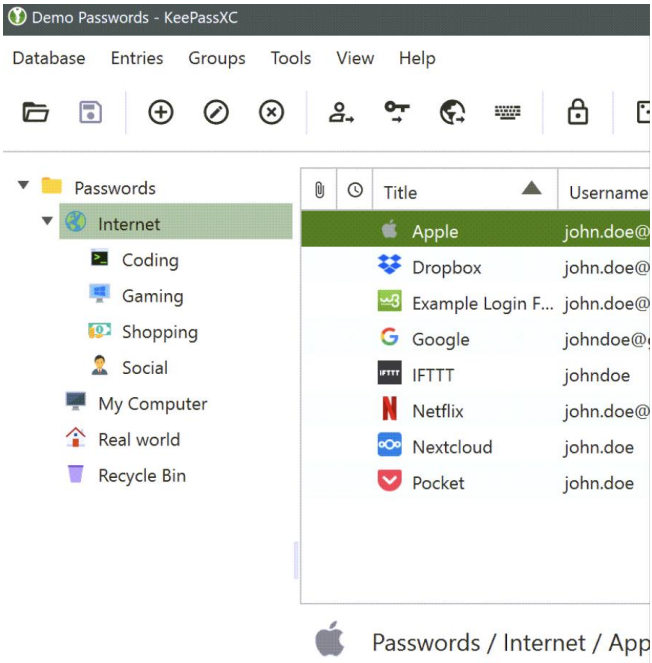
г) автономність (працює локально без потреби у підключенні до Інтернету, що підвищує безпеку).

Програму можна завантажити з офіційного сайту, вона вільна, безкоштовна і має відкритий код (для розробників) https://keepassxc.org. Окрім згаданих платформ існує ще додаток до браузера. Програма має розширену документацію й інструкції (https://keepassxc.org/docs/KeePassXC_GettingStarted) .

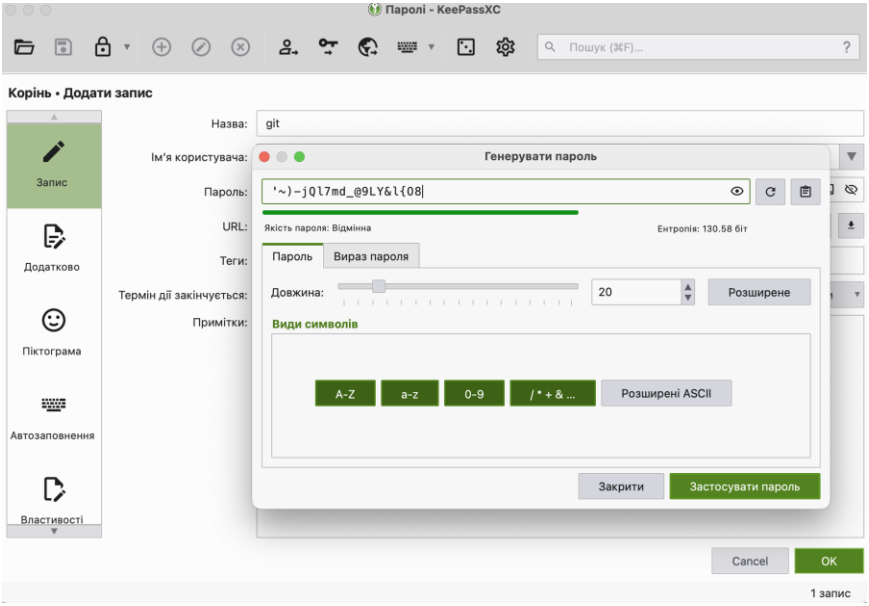
Після встановлення і запуску програми буде пропоновано або відкрити базу паролів, яка існує, або створити нову та зберегти її у вказаній теці (як показано на рисунку)



У програмі можна створювати, групувати паролі (разом з іменем користувача), включати/виключати інтеграції з браузером



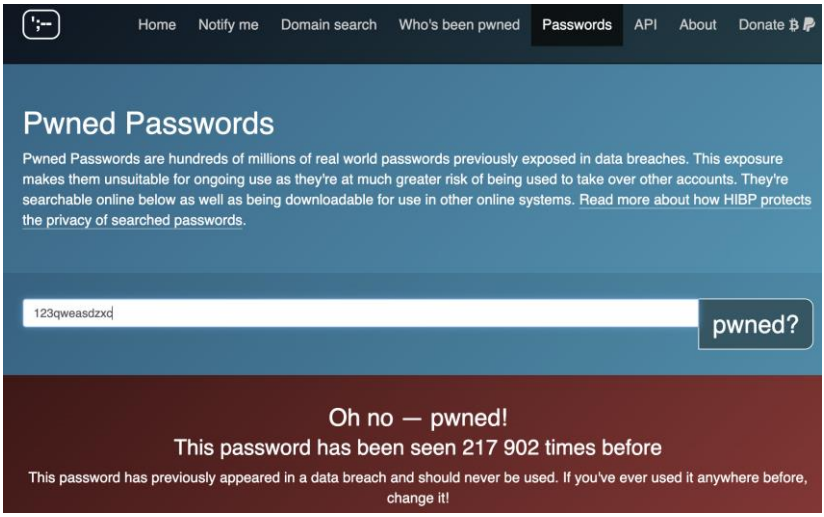
При створенні паролю, можна скористатися ефективним генератором паролів, який дає змогу вибрати набір символів (які будуть використовуватися), довжину пароля, вираз для пароля (досить часто для пароля використовують довгу фразу з специфічним розділювачем). При цьому система автоматично відзначить рівень надійності пароля.



Також слід відзначити, KeePassXC не дозволяє по замовчуванні роботи скрин вікна. Щоб робити скрини вікна програму слід запустити з прапорцем `**--allow-screencapture**`.

Досить часто після створення паролю важливо перевірити чи він не є скомпрометованим. Перевірка паролів за базою скомпрометованих паролів, такою як **Pwned Passwords** на сайті [**Have I Been Pwned**](https://haveibeenpwned.com/Passwords), дозволяє користувачам дізнатися, чи їхній пароль було скомпрометовано у випадку витоку даних. Це допомагає уникнути використання небезпечних паролів, які вже є в базах даних зламаних облікових записів. Pwned Passwords використовує алгоритм **k-Anonymity**, щоб захистити конфіденційність паролів під час перевірки.

Ось як це працює. Спочатку введений пароль хешується за допомогою алгоритму SHA-1. Наприклад, якщо ви вводите пароль "`password`", він перетворюється на хеш `5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8`. Для перевірки до сервера надсилається лише перші 5 символів хешу. У нашому прикладі це буде `5baa6`. По базах з сервера повертається список усіх хешів, які починаються з цих 5 символів, разом із кількістю разів, коли кожен пароль було знайдено в базі. Клієнтська сторона порівнює хеш пароля з отриманими результатами. Таким чином, сервер ніколи не отримує повний хеш або сам пароль. Завдяки використанню методу k-Anonymity повний пароль або його хеш залишається локальним на вашому пристрої. Це забезпечує високий рівень конфіденційності та безпеки. У KeePassXC також є вбудована функція перевірки паролів через Pwned Passwords. Вона використовує той самий принцип k-Anonymity, що й офіційний сервіс, забезпечуючи конфіденційність і запобігаючи витоку



ваших паролів. Якщо ж ваш пароль виявлено в базі скомпрометованих, негайно змініть його на всіх сервісах, де він використовується.

Як видно з рисунка, пароль `123qweasdzxc`, згідно з базами сервісу, був використаний 217902 разів. Це робить його не надійним. А ось пароль `123qweasdzxcrfvtgbyhn!` на момент написання цієї лабораторної роботи не був скомпрометований (хоча його оприлюднення у цій книзі компрометує його).

Таким чином, перевірка паролів через Pwned Passwords є безпечним і корисним інструментом для підвищення рівня вашої інформаційної безпеки.

Google Chrome пропонує вбудовану функцію збереження паролів, яка допомагає автоматизувати процес входу на сайти. Цей менеджер паролів є зручним для багатьох користувачів, оскільки дозволяє швидко зберігати та використовувати паролі. Збереження паролів у Chrome працює так

1. Коли ви вперше входите на сайт, Chrome автоматично пропонує зберегти пароль.

2. Збережені паролі зберігаються у вашому обліковому записі Google, якщо ви увійшли в Chrome. Ви можете переглянути їх у налаштуваннях браузера або на сторінці [**Google Password Manager**](https://passwords.google.com).

3. Chrome автоматично підставляє збережений логін і пароль при наступному відвідуванні сайту.

Ці функції паролів у Chrome забезпечують швидкий доступ до облікових записів без необхідності запам'ятовувати паролі. Також ці паролі синхронізуються між усіма пристроями, де ви використовуєте Chrome і увійшли в той самий обліковий запис Google. А вбудований менеджер паролів дозволяє легко

керувати збереженими даними. Щоб зберегти паролі у Chrome треба:

1. Відкрийте Chrome і увійдіть у свій обліковий запис Google.

2. Увімкніть опцію збереження паролів:

- Перейдіть у `Settings` → `Autofill` → `Passwords`.

- Увімкніть "Offer to save passwords" (Пропонувати зберігати паролі).

3. Введіть дані для входу на сайті, і Chrome запропонує зберегти пароль. Натисніть "Save" (Зберегти).

Однак збереження паролів у Chrome має і недоліки:

1. Ризик компрометації (якщо зловмисник отримає доступ до вашого облікового запису Google, він також матиме доступ до ваших паролів). Цей ризик можна зменшити використовуючи двофакторну авторизацію.

2. Відсутність складніших функцій (менеджер паролів у Chrome не пропонує таких просунутих можливостей, як категоризація або підтримка двофакторної аутентифікації).

3. Локальна небезпека (якщо ваш комп’ютер не захищений паролем або шифруванням, зловмисник може отримати доступ до збережених паролів.

Щоб переглянути збережені паролі у Chrome

1. Відкрийте Chrome і перейдіть у `Settings` → `Autofill` → `Passwords`.

2. У розділі "Saved Passwords" ви побачите перелік сайтів і збережених облікових даних.

3. Щоб переглянути пароль, натисніть на іконку ока та введіть пароль від вашого комп'ютера.

Таким чином, збереження паролів у Chrome є зручним рішенням для щоденного використання, але для критично

важливих облікових записів краще використовувати окремий менеджер паролів для максимального захисту.

Завдання

1. Увійдіть у власний обліковий запис Google і перевірте 1-2 із збережених паролів на предмет компрометації.

2. Запустіть KeePassXC і створіть власну базу паролів.

3. Згенеруйте декілька паролів за допомогою KeePassXC з використанням різних символів і текстів.

4. Перевірте надійність паролів зі списку. Оцініть їх за критеріями: надійність, запам’ятовуваність, компрометація. Складіть таблицю. Список паролів:

```
qwerty12345
P@ssw0rd!2025
1234567890
password1
Letmein2025!
aB1cdEf2G!
123qwe!@#
XxX!2025secureXxX
admin2023
qazwsx@2025
Y0u!rock2019
hello12345
S3curePassword!2025
Monkey123!
1qaz2wsx3edc
QwErTy!1
zxcvbnm098
MySecurePass_2025
```

```
abcdefg12345
p@ssw0rD!!
```

5. Використовуйте Pwned Passwords для перевірки надійності паролів. Для цього зайдіть на сайт Have I Been Pwned і введіть кілька паролі зі списку для перевірки на компрометацію. Запишіть також у таблицю результати для кожного пароля: скільки разів він був знайдений у базах даних зламаних акаунтів.

6. Перевірте те саме (пункт 5) через KeePassXC. Додайте до таблиці інформацію.

7. Оформіть звіти і захистіть лабораторну роботу.

Контрольні питання

1. Що таке пароль і для чого він використовується?

2. Які основні вимоги до безпеки паролів?

3. Чому важливо використовувати унікальні паролі для різних облікових записів?

4. Як складність пароля впливає на його безпеку?

5. Які типи символів повинні бути включені в складний пароль?

6. Чому паролі не повинні зберігатися у відкритому вигляді?

7. Які переваги використання менеджерів паролів для їх збереження?

8. Що таке шифрування паролів і як воно підвищує безпеку?

9. Які алгоритми шифрування використовуються в сучасних менеджерах паролів?

10. Чому KeePassXC є безпечним рішенням для збереження паролів?

11. Як KeePassXC захищає дані в базі паролів?

12. Чим KeePassXC відрізняється від інших менеджерів паролів?

13. Як згенерувати надійний пароль за допомогою KeePassXC?

14. Що таке Pwned Passwords і як він допомагає перевірити безпеку пароля?

15. Як працює алгоритм k-Anonymity для перевірки паролів?

16. Як перевірити, чи потрапив пароль у базу скомпрометованих?

17. Які переваги та недоліки перевірки паролів через Pwned Passwords?

18. Як забезпечити конфіденційність під час перевірки паролів за допомогою Pwned Passwords?

19. Чим відрізняється перевірка паролів через Pwned Passwords у KeePassXC від перевірки на сайті?

20. Як створити нову базу паролів у KeePassXC?

21. Як організувати паролі в KeePassXC за категоріями?

22. Як включити інтеграцію KeePassXC з браузером?

23. Як налаштувати KeePassXC для роботи з генератором паролів?

24. Чому KeePassXC не дозволяє робити скріншоти вікна за замовчуванням і як це змінити?

25. Як зберігати паролі у браузері Google Chrome?

26. Які переваги збереження паролів у Google Chrome?

27. Які недоліки збереження паролів у Google Chrome?

28. Як здійснити синхронізацію паролів між пристроями за допомогою Google Chrome?

29. Як перевірити наявність збережених паролів у Google Chrome?

30. Чому для критичних облікових записів краще використовувати менеджери паролів замість вбудованих функцій браузера?

Лабораторна робота № 2. Створення резервної копії даних і шифрування файлів

Мета: Навчитися створювати резервні копії даних для забезпечення їх збереження, а також освоїти методи шифрування файлів для захисту конфіденційної інформації від несанкціонованого доступу.

Обладнання: Персональні комп'ютери з встановленою операційною системою (Windows/Linux/macOS).

Програмне забезпечення: програма-менеджер Google Drive for Desktop, програма Cryptomator .

Теоретичні відомості

Створення резервних копій даних є важливим для забезпечення їх збереження і доступності у разі непередбачуваних ситуацій. У сучасному цифровому світі інформація часто є одним із найцінніших ресурсів, а її втрата може призвести до значних проблем як для окремих користувачів, так і для бізнесу. Резервні копії слугують своєрідною страховкою від апаратних збоїв, програмних помилок, вірусів, кібератак чи випадкового видалення файлів.

Основна мета створення резервних копій – гарантувати відновлення важливих даних у разі їхньої втрати. Наприклад, у випадку виходу з ладу жорсткого диска чи інших пристроїв зберігання інформації, резервна копія дозволяє швидко відновити доступ до потрібних файлів. Вона також відіграє важливу роль у захисті від програм-вимагачів, які можуть заблокувати дані на комп'ютері, вимагаючи викуп. Завдяки резервному копіюванню користувачі можуть обійтися без виконання таких вимог і відновити доступ до своїх файлів.

Крім того, резервні копії є критично важливими для підприємств і організацій. Втрата корпоративних даних, таких як фінансові записи, інформація про клієнтів чи стратегічні документи, може завдати серйозної шкоди репутації компанії, спричинити фінансові втрати або навіть призвести до закриття бізнесу. Для багатьох організацій резервне копіювання є обов'язковою практикою, яка забезпечує безперервність бізнес-процесів.

Для індивідуальних користувачів резервне копіювання дозволяє зберегти сімейні фотографії, особисті документи чи улюблені мультимедійні файли, які часто мають неоціненну емоційну чи культурну цінність. Завдяки сучасним технологіям резервні копії можна створювати автоматично, використовуючи хмарні сервіси або локальні пристрої, що робить цей процес простим і доступним для кожного.

Найбільш поширеною, доступною та багатоплатформовою є програма **Google Drive for Desktop**. Google Drive for Desktop – це інструмент для синхронізації файлів між хмарним сховищем **Google Drive** та локальним комп’ютером. Він дозволяє користувачам отримувати доступ до файлів безпосередньо з комп’ютера без необхідності відкривати веббраузер. Програма підтримує як потокове відображення файлів, коли дані завантажуються лише за потреби, так і повне синхронізування вибраних тек для доступу в режимі офлайн. Користувачі можуть налаштовувати, які файли чи теки зберігати локально, а які залишити в хмарі. Google Drive for Desktop інтегрується з іншими інструментами Google, наприклад, **Google Docs**, **Sheets** і **Slides**, дозволяючи відкривати, редагувати та автоматично зберігати зміни в хмарі. Крім цього, підтримується синхронізація з **Google Photos**, що полегшує організацію та резервне

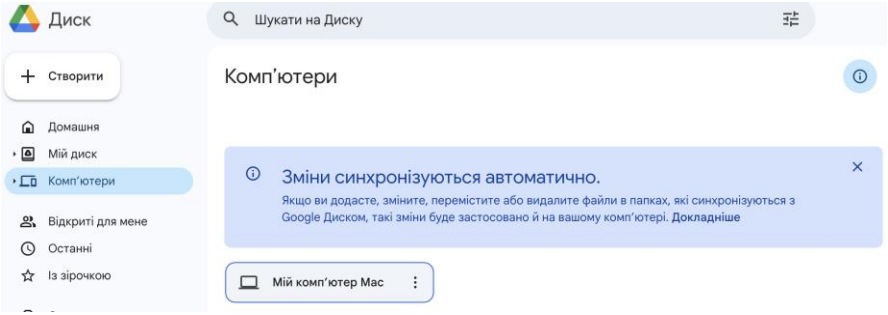
копіювання фотографій. Інструмент також забезпечує простий обмін файлами за допомогою посилань та підтримує багатокористувацьке середовище з можливістю доступу до декількох облікових записів Google.

Резервне копіювання можна налаштовувати вручну і копіювати дані на гугл диск самому. Однак Google Drive for Desktop має можливість синхронізації потрібних тет з Google Drive.

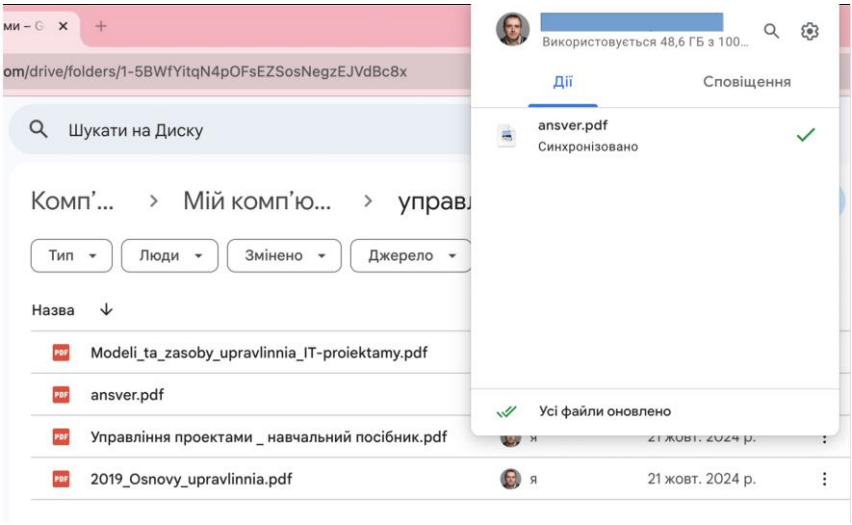
Щоб організувати синхронізацію відкрите налаштування Google Drive for Desktop → клацніть на іконку Google Drive (в області повідомлень у Windows або на панелі у macOS) →та натисніть на значок ⚙ (шестерня) і виберіть "Налаштування" → перейдіть до розділу "Мій диск" → у налаштуваннях знайдіть розділ "Мій диск" або "Google Drive" і налаштуйте синхронізацію. Оберіть опцію "Потокове передавання файлів" (файли залишатимуться в хмарі, але доступні при необхідності) або "Синхронізувати файли на цей комп'ютер" (файли будуть завантажені локально). Натисніть "Синхронізувати тільки вибрані теки". Позначте лише ту теку, яку ви хочете синхронізувати, і зніміть галочки з інших.

Програма також має можливість налаштувати резервне копіювання. У налаштуваннях Google Drive виберіть розділ "Додати теку для резервного копіювання". Далі натисніть "Додати теку", виберіть потрібну на комп'ютері та налаштуйте синхронізацію з Google Drive. Потім підтвердіть вибір, обравши, чи синхронізувати файли тільки з Google Drive або також із Google Photos.

Коли синхронізація організована, то на гугл диску можна побачити перелік комп’ютерів, які синхронізуються.



Якщо відкрити вибраний комп’ютер, то видно список тек. Якщо туди додати файл, то він одразу появиться як локально на вибраному комп’ютер (якщо комп’ютер включений, є доступ в інтернет і включена синхронізація) так і в хмарі Google Drive. Це видно на скрині нижче

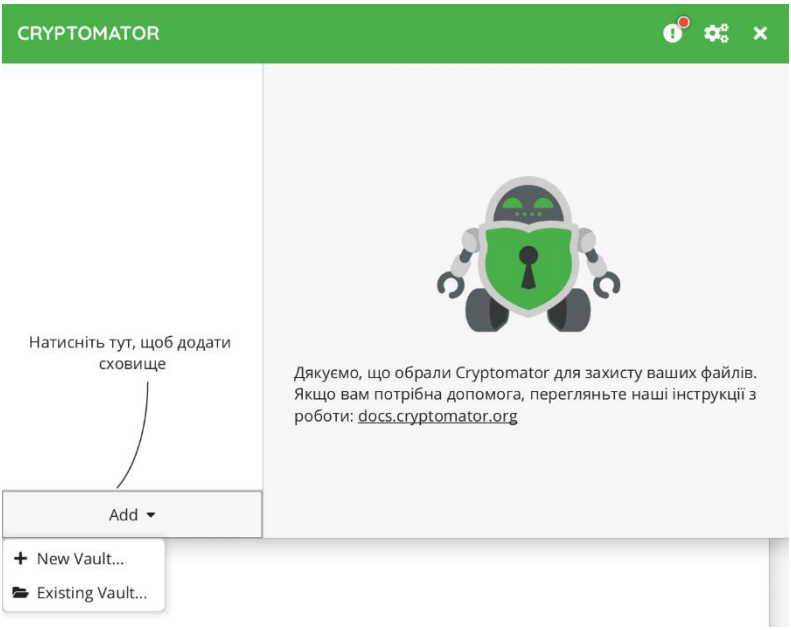


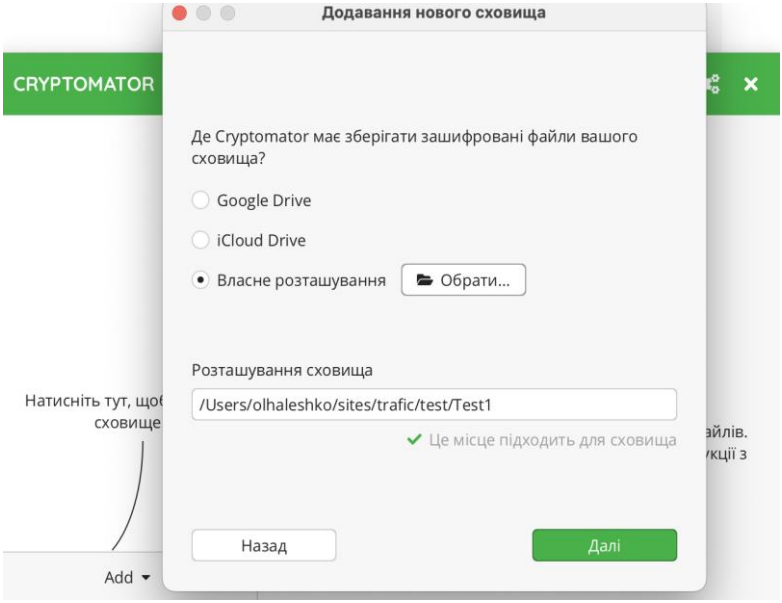
Створення резервної копії — це важливий крок для збереження ваших даних, але саме по собі це лише перша лінія захисту. Убезпечення резервної копії та інших файлів, особливо в хмарних сховищах, потребує додаткового рівня безпеки. І саме тут у гру вступає **Cryptomator**.

Cryptomator — це простий у використанні, безкоштовний і відкритий інструмент для шифрування ваших даних перед їх завантаженням у хмару. Він створює віртуальний сейф (або сховище), в якому всі файли автоматично шифруються. Навіть якщо хтось отримає доступ до вашого хмарного акаунта, файли залишаться захищеними, адже без пароля або ключа шифрування вони будуть виглядати як набір безглуздих символів. Cryptomator забезпечує кінцеве шифрування (end-to-end encryption): дані шифруються на вашому пристрої перед передачею в хмару. Він має простий у використанні інтерфейс, немає необхідності мати технічні знання — створення та управління сейфами інтуїтивно зрозуміле. Cryptomator сумісний із популярними хмарами: Google Drive, Dropbox, OneDrive та іншими. І він реалізований на різних платформах: Windows, macOS, Linux, iOS та Android.

Використання Cryptomator гарантує, що дані захищені від несанкціонованого доступу, навіть у випадку злому акаунта чи компрометації хмарного провайдера.

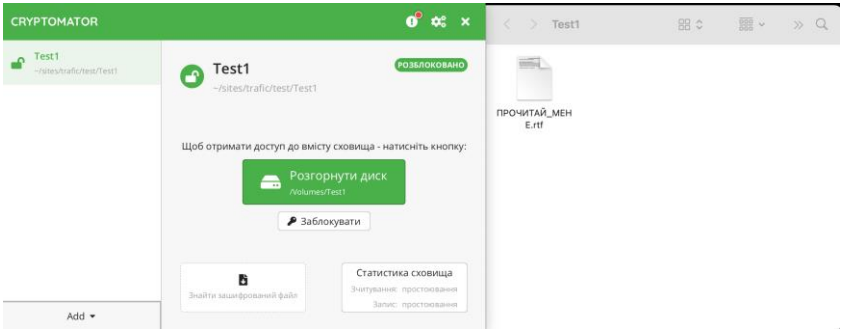
При запуску програми пропонується створення нового сховища


CRYPTOMATOR
Натисніть тут, щоб додати сховище
Add
New Vault...
Existing Vault...
Дякуємо, що обрали Cryptomator для захисту ваших файлів. Якщо вам потрібна допомога, перегляньте наші інструкції з роботи: docs.cryptomator.org


CRYPTOMATOR
Додавання нового сховища
Де Cryptomator має зберігати зашифровані файли вашого сховища?
Google Drive
iCloud Drive
Власне розташування
Обрати...
Розташування сховища
/Users/olhaleshko/sites/trafic/test/Test1
Це місце підходить для сховища
Назад
Далі
Add

Вибираємо нове значення і слідуємо майстру налаштувань. Тут вказуємо назву сховища, і вказуємо, де воно буде розташоване. Тут можна вказати як локальне розташування, так і хмарне. Вибравши розташування сховища, задаємо пароль і створюємо сховище.

Для доступу до сховища слід ввести пароль і розблокувати його.



Після цього сховище доступне і в нього можна додавати файли, теки і працювати у звичайному режимі. Якщо ж сховище заблокувати, то доступ до нього закривається, і всі документи у ньому залишаються зашифровані і незрозумілі для читання.

Завдання

1. Налаштуйте Cryptomator на своєму комп’ютері. Додайте теку для шифрування.

2. Створіть інші сховища з різними варанами збереження даних (локально і у хмарі).

3. Налаштуйте Google Drive for Desktop.

4. Синхронізуйте сховище, що зроблено в Cryptomator з Google Drive.

5. Оформіть звіти і захистіть лабораторну роботу.

Контрольні питання

1. Що таке резервне копіювання і для чого воно потрібне?

2. Чому втрата даних може бути критичною як для окремих користувачів, так і для бізнесу?

3. Які основні причини створення резервних копій?

4. Як резервне копіювання допомагає захиститися від програм-вимагачів?

5. Чому резервні копії важливі для підприємств і організацій?

6. Які наслідки можуть виникнути у разі втрати корпоративних даних?

7. Яку роль відіграють резервні копії для індивідуальних користувачів?

8. Як хмарні сервіси спрощують процес створення резервних копій?

9. Що таке Google Drive for Desktop і для чого його використовують?

10. Які способи синхронізації файлів підтримує Google Drive for Desktop?

11. Як налаштувати синхронізацію вибраних тек у Google Drive for Desktop?

12. У чому різниця між потоковим передаванням файлів і повною синхронізацією?

13. Які переваги інтеграції Google Drive for Desktop з іншими інструментами Google?

14. Як налаштувати резервне копіювання у Google Drive for Desktop?

15. Що потрібно для початку роботи з Google Drive for Desktop?

16. Як перевірити перелік комп'ютерів, синхронізованих з Google Drive?

17. Що відбувається з файлом, доданим у синхронізовану теку?

18. Чому створення резервної копії – це лише перший крок у захисті даних?

19. Що таке Cryptomator і яку функцію він виконує?

20. Як Cryptomator забезпечує шифрування даних?

21. Які особливості роботи з Cryptomator виділяють його серед інших інструментів?

22. Чому Cryptomator є ефективним для захисту даних у хмарних сховищах?

23. Як створити нове сховище у Cryptomator?

24. Які платформи підтримує Cryptomator?

25. Як задати пароль для нового сховища у Cryptomator?

26. Що потрібно зробити для доступу до сховища в Cryptomator?

27. Що відбувається з файлами у сховищі після його блокування?

28. Як вибрати розташування сховища у Cryptomator (локальне або хмарне)?

29. У чому полягає принцип роботи кінцевого шифрування (end-to-end encryption)?

30. Чому важливо шифрувати дані навіть у випадку використання хмарних провайдерів?

Лабораторна робота № 3. Багатофакторна автентифікація

Мета: Ознайомитися з принципами багатофакторної авторизації, дослідити її основні механізми та реалізувати її для підвищення безпеки доступу до інформаційних ресурсів.

Обладнання: Персональні комп'ютери з встановленою операційною системою (Windows/Linux/macOS) із доступом до Інтернету; смартфон із встановленими додатками.

Програмне забезпечення: програма Google Authenticator.

Теоретичні відомості

Багатофакторна автентифікація (БФА) є одним із найефективніших методів забезпечення безпеки доступу до інформаційних систем та ресурсів. Її принцип базується на використанні декількох незалежних факторів для підтвердження особи користувача. Мета цієї технології – знизити ризики несанкціонованого доступу навіть у разі компрометації одного з методів аутентифікації.

Фактори аутентифікації зазвичай поділяють на три основні категорії: те, що знає користувач (наприклад, пароль або PIN-код), те, що належить користувачу (наприклад, смартфон, смарт-карта або апаратний ключ безпеки), та те, ким є користувач (біометричні дані, такі як відбиток пальця або розпізнавання обличчя). Ключовим принципом БФА є вимога використання щонайменше двох факторів із різних категорій для забезпечення доступу.

БФА отримала широке розповсюдження завдяки зростанню кількості кіберзагроз і витоків даних. У багатьох випадках введення лише пароля є недостатнім, оскільки паролі

можуть бути викрадені або зламані. Додавання додаткових факторів значно ускладнює завдання зловмисникам, оскільки для доступу їм необхідно скомпрометувати всі задіяні механізми.

Найбільш поширеними прикладами багатофакторної автентифікації є комбінація пароля та одноразового коду, який генерується мобільним додатком або надсилається через SMS. Інші популярні методи включають використання апаратних ключів безпеки, що підтримують стандарти, такі як FIDO, або біометричні дані, інтегровані у смартфони чи ноутбуки. Завдяки цим технологіям користувачі можуть безпечно входити в облікові записи навіть у випадках, коли їхні паролі стають відомими третім особам.

БФА активно впроваджується у фінансових установах, корпоративних системах, соціальних мережах та інших сервісах, де безпека даних має критичне значення. Окрім забезпечення конфіденційності інформації, багатофакторна автентифікація також підвищує довіру користувачів до сервісу, що використовує такі заходи безпеки.

Google пропонує зручний та надійний механізм БФА, який включає одноразові коди, апаратні ключі безпеки або сповіщення через додаток Google. Налаштування можна виконати, дотримуючись наступних кроків:

1. Увійдіть до свого облікового запису Google. Відкрийте [Google Аккаунт](https://myaccount.google.com/) та увійдіть, використовуючи свій логін і пароль.

2. Перейдіть до налаштувань безпеки. У меню зліва виберіть розділ "Безпека".

3. Активуйте двоетапну перевірку. У розділі "Увійти в обліковий запис Google" знайдіть пункт "Двоетапна перевірка" та натисніть "Увімкнути".

4. Виберіть метод перевірки. Google запропонує кілька варіантів:

а) сповіщення на вашому телефоні (Google Prompt);

б) текстове повідомлення (SMS) або телефонний дзвінок;

в) використання Google Authenticator для генерації одноразових кодів;

г) апаратний ключ безпеки (за наявності).

5. Налаштуйте вибраний метод. Наприклад, якщо ви обираєте додаток Google Authenticator:

а) завантажте додаток із App Store або Google Play;

б) скануйте QR-код, запропонований Google, або введіть код вручну.

6. Завершіть налаштування. Після налаштування Google попросить ввести одноразовий код або підтвердити дію через обраний метод, щоб завершити процес.

7. Додайте резервний метод. Ви можете налаштувати резервні коди, які можна використовувати у разі втрати доступу до основного пристрою. Збережіть їх у безпечному місці.

Розгленемо ще один приклад налаштування багатофакторної автентифікації (БФА) у Facebook. Виконайте такі кроки:

1. Увійдіть до свого акаунта Facebook. Відкрийте [Facebook](https://www.facebook.com/) і ввійдіть за допомогою логіна і пароля.

2. Перейдіть до налаштувань безпеки. Натисніть на значок меню (три горизонтальні лінії або стрілка вниз) у верхньому

правому куті. Оберіть "Налаштування та конфіденційність", а потім "Налаштування". Перейдіть у розділ "Безпека та вхід".

3. Увімкніть двофакторну автентифікацію. Знайдіть розділ "Двофакторна автентифікація" та натисніть "Редагувати".

4. Виберіть метод автентифікації. Facebook запропонує кілька варіантів:

а) використання програми для автентифікації (наприклад, Google Authenticator або Authy);

б) надсилання кодів через SMS;

в) апаратний ключ безпеки (за наявності).

5. Налаштуйте вибраний метод. Наприклад, для програми-генератора кодів:

а) завантажте додаток на ваш смартфон;

б) відскануйте QR-код, запропонований Facebook, або введіть код вручну;

6. Перевірте налаштування. Введіть код із додатку або SMS, щоб підтвердити успішність налаштування.

7. Додайте резервні опції. Налаштуйте резервні методи, такі як:

а) резервні коди, які можна використовувати у випадку втрати доступу до основного методу;

б) призначення довіреного пристрою або контактів.

Таким чином, налаштувавши багатофакторну автентифікацію у Google та Facebook, ви зможете значно підвищити безпеку своїх облікових записів. Аналогічним чином налаштовується багатофакторна автентифікація на інших популярних платформах.

Завдання

1. Перевірте, чи підтримує ваш акаунт Google багатофакторну автентифікацію, і налаштуйте її, вибравши один із доступних методів (наприклад, Google Authenticator або SMS-коди).

2. Налаштуйте багатофакторну автентифікацію у своєму акаунті Facebook, вибравши метод автентифікації за допомогою програми-генератора кодів.

3. Перевірити працездатність налаштувань, увійшовши до акаунта через інший пристрій або браузер, використовуючи новий метод автентифікації.

4. Розгляньте інші сервіси, що підтримують БФА (Instagram, Dropbox,Twitter (X), LinkedIn, GitHub,Pinterest), налаштуйте БФА на них і опишіть зі скринами налаштування і результат.

5. Оформіть звіти і захистіть лабораторну роботу.

Контрольні питання

1. Що таке багатофакторна автентифікація (БФА)?

2. Яка основна мета використання багатофакторної автентифікації?

3. Які основні категорії факторів автентифікації існують?

4. Наведіть приклади факторів автентифікації для категорії "те, що знає користувач".

5. Наведіть приклади факторів автентифікації для категорії "те, що належить користувачу".

6. Чому використання лише пароля є недостатньо безпечним?

7. Як багатофакторна автентифікація підвищує безпеку облікових записів?

8. Які найпоширеніші методи багатофакторної автентифікації використовуються?

9. Що таке Google Authenticator?

10. Як налаштувати двоетапну перевірку в обліковому записі Google?

11. Які методи перевірки пропонує Google для БФА?

12. Що таке резервні коди, і чому їх важливо зберігати в безпечному місці?

13. Які ризики можуть виникнути за відсутності багатофакторної автентифікації?

14. Як налаштувати багатофакторну автентифікацію у Facebook?

15. Які методи БФА підтримує Facebook?

16. Як використовувати програму-генератор кодів для автентифікації?

17. Чому SMS-повідомлення не є найкращим методом для БФА?

18. Що таке апаратний ключ безпеки, і як він працює?

19. Які переваги біометричних методів автентифікації?

20. У яких сферах багатофакторна автентифікація використовується найчастіше?

21. Чому БФА є критично важливою для фінансових установ?

22. Як багатофакторна автентифікація впливає на довіру користувачів до сервісу?

23. Назвіть три хмарні сервіси, які підтримують БФА.

24. Які популярні соціальні мережі підтримують БФА?

25. Які дії слід виконати, якщо втрачено доступ до пристрою, на якому налаштовано БФА?

26. Як перевірити, чи підтримує ваш обліковий запис БФА?

27. Які проблеми можуть виникнути під час налаштування БФА, і як їх вирішити?

28. Чим відрізняється Google Authenticator від інших програм-генераторів кодів?

29. Як налаштувати БФА на іншому популярному сервісі (наведіть приклад)?

30. Як багатофакторна автентифікація допомагає запобігти фішинг-атакам?

Лабораторна робота № 4. Основи електронного підпису

Мета: Вивчити і засвоїти практичні методи створення та верифікації простого електронного підпису з використанням відповідних інструментів та програмного забезпечення.

Обладнання: Персональні комп'ютери з встановленою операційною системою (Windows/Linux/macOS).

Програмне забезпечення: OpenSSL (для генерації ключів, підписання та верифікації), текстовий редактор (для створення та перегляду файлів), веб-браузер (для доступу до документації та онлайн-ресурсів).

Теоретичні відомості

Простий електронний підпис (**ПЕП**) – це електронні дані, що додаються до інших електронних даних або логічно з ними пов'язані та призначені для ідентифікації підписувача. Він відрізняється від кваліфікованого електронного підпису (**КЕП**) тим, що не потребує використання кваліфікованого сертифіката від акредитованого центру сертифікації ключів. Це робить його простішим у використанні, але й менш захищеним.

ПЕП потрібний для швидкої та зручної ідентифікації особи в електронному середовищі, де не потрібен високий рівень захисту. Він часто використовується для внутрішнього документообігу в організаціях, для підтвердження згоди з умовами на веб-сайтах, для авторизації на онлайн-платформах, для підписання некритичних документів, таких як внутрішні накази, заявки, листування тощо.

Важливість ПЕП полягає у спрощенні електронної взаємодії. Він дозволяє швидко підтвердити особу відправника

або факт згоди з певними умовами без складних процедур отримання та використання КЕП. Це сприяє підвищенню ефективності роботи та зручності користування електронними сервісами. Однак, важливо розуміти, що ПЕП не гарантує на 100% цілісність підписаного документа та не забезпечує надійний захист від підробки. Тому його не рекомендується використовувати для підписання юридично значущих документів, фінансових транзакцій або інших операцій, де потрібна висока ступінь довіри та захисту. У таких випадках необхідно використовувати КЕП.

Отже, ПЕП є корисним інструментом для швидкої ідентифікації та підтвердження дій в електронному середовищі, але його використання має бути обмежене сферами, де низький рівень ризику та не потрібен високий рівень юридичної значущості.

Розглянемо можливість створення і підписання документів за допомогою **OpenSSL** – потужного та широко поширеного інструменту з відкритим вихідним кодом, який надає бібліотеки для криптографічних операцій. Він реалізує протоколи Secure Sockets Layer (SSL) та Transport Layer Security (TLS), які забезпечують безпечну передачу даних в мережі, зокрема в інтернеті. OpenSSL є фактичним стандартом у світі криптографії та використовується в багатьох веб-серверах, додатках та інших системах. До важливих криптографічних функцій OpenSSL можна віднести:

1) симетричне шифрування (AES, DES, Blowfish тощо);
2) асиметричне шифрування (RSA, DSA, ECC);
3) хешування (MD5, SHA1, SHA2, SHA3)
4) цифрові підписи;
5) генерація ключів (приватних та публічних);

6) підписання та перевірка цифрових підписів;

7) шифрування та розшифрування даних;

8) керування сертифікатами.

Також слід відзначити, що OpenSSL працює на різних операційних системах, таких як Linux, Windows, macOS та інші.

Розглянемо використання OpenSSL для накладання ПЕП на документи. Для цього слід спочатку згенерувати пару ключів: приватний та публічний. Приватний ключ зберігається в секреті та використовується для підписання. Публічний ключ розповсюджується відкрито та використовується для перевірки підпису. Це основна ідея асиметричного шифрування. OpenSSL використовує алгоритм **RSA** – один з найпоширеніших алгоритмів асиметричного шифрування для генерації ключів; це криптографічний алгоритм з відкритим ключем, що базується на складності факторизації великих чисел. Отже, для генерації приватного ключа слід виконати команду в PowerShell (під Windows) чи в bash-терміналі (під Linux) чи zsh-терміналі (під Mac):

```
openssl genrsa -out private.pem 2048
```

У результаті буде створено файл-приватний ключ `**private.pem**`. Для створення публічного ключа слід виконати команду:

```
openssl rsa -in private.pem -pubout -out public.pem
```

У результаті буде створено файл-публічний ключ `**public.pem**`. Ці файли будуть використовуватися для підписання документів і перевірки підпису.

Розглянемо процес підписання pdf-документа `**my_doc.pdf**`. Для цього слід обидва ключі й потрібний документ помітити в одну теку і виконати команду:

```
openssl dgst -sha256 -sign private.pem -out signature.bin my_doc.pdf
```

Тут опція `**dgst**` вказує OpenSSL виконати операцію дайджесту (хешування) та/або цифрового підпису. Дайджест – це короткий "відбиток" даних, отриманий за допомогою хеш-функції. Опція `**-sha256**` визначає алгоритм хешування, який буде використано для створення дайджесту файлу. `**sha256**` означає використання алгоритму SHA-256 (Secure Hash Algorithm 256-bit). SHA-256 є криптографічно стійким алгоритмом хешування, який генерує 256-бітний (32-байтний) хеш. Важливо використовувати саме стійкі алгоритми, оскільки старіші, такі як MD5 або SHA1, вважаються вразливими. Опція `**-sign private.pem**` вказує OpenSSL виконати операцію цифрового підпису. `**private.pem**` – це ім'я файлу, що містить приватний ключ у форматі PEM. *Важливо ніколи не передавати свій приватний ключ нікому! Він повинен зберігатися в безпечному місці.* Опція `**-out signature.bin**` визначає ім'я файлу, в який буде збережено створений цифровий підпис. `**signature.bin**` – це і є ім'я вихідного файлу. Розширення `**.bin**` вказує на те, що це бінарний файл. `**my_doc.pdf**` – це ім'я файлу, який слід підписати. OpenSSL обчислить хеш цього файлу, а потім зашифрує цей хеш за допомогою приватного ключа.

Для того, щоб інший користувач міг перевірити підписаний документ, йому потрібно передати наступні файли:

1) сам підписаний документ `my_doc.pdf`. Це оригінальний файл, який був підписаний. Без нього перевірка підпису неможлива;

2) файл підпису `signature.bin`. Цей файл містить цифровий підпис, створений за допомогою вашого приватного ключа. Він є результатом виконання команди з підписання.

3) публічний ключ public.pem. Цей ключ відповідає вашому приватному ключу private.pem, яким було створено підпис. Публічний ключ використовується для розшифрування підпису та порівняння хешів. Публічний ключ ***можна передавати відкрито***, оскільки він не містить жодної секретної інформації. ***А приватний ключ private.pem не слід передавати нікому! Він повинен залишатися в безпеці та доступний лише вам. Передача приватного ключа рівносильна втраті контролю над вашими підписами.***

Отже, після підписання усі 3 файли можна надсилати отримувачу, який зможе перевірити чи `my_doc.pdf` підписаний саме тією особою, яка передає документ. Для перевірки слід виконати команду:

```
openssl dgst -sha256 -sign private.pem -out signature.bin my_doc.pdf
```

Якщо підпис дійсний, OpenSSL виведе повідомлення `**Verified OK**`. Це означає, що файл `my_doc.pdf` не було змінено після підписання, і підпис було створено за допомогою приватного ключа, відповідного наданому публічному ключу. Якщо підпис недійсний, OpenSSL виведе повідомлення `**Verification Failure**`. Це означає, що або Файл `my_doc.pdf` було змінено після підписання, або використано неправильний публічний ключ; або файл підпису signature.bin пошкоджено.

Завдання

1. Згенеруйте пару публічний і приватний ключі у своїй робочій теці за допомогою OpenSSl.

2. Створіть довільний текстовий чи pdf-документ.

3. Підпишіть документ.

4. Надішліть поштою одному з ваших одногрупників 3 потрібні файли (документ_1, файл підпису, публічний ключ) для ідентифікації вашого підпису в підписаному документі. До листа також додайте ще один документ (непідписаний), документ_2.

5. Отримайте від одногрупника лист з 4-ма файлами і визначіть який документ підписано, а який ні.

6. Отримайте від викладача лист з

1) переліком публічних ключів і їх авторів,

2) переліком файл підписів,

3) переліком документів.

7. Визначіть які документи і ким підписані.

8. Оформіть звіти і здайте їх.

Контрольні питання

1. Що таке простий електронний підпис (ПЕП)?

2. Чим ПЕП відрізняється від кваліфікованого електронного підпису?

3. Для чого використовується ПЕП?

4. Де часто застосовують ПЕП?

5. Які переваги використання ПЕП?

6. Які недоліки ПЕП?

7. Чи гарантує ПЕП 100% цілісність документа?

8. Чи забезпечує ПЕП надійний захист від підробки?

9. В яких випадках не рекомендується використовувати ПЕП?

10. Яке програмне забезпечення використовується в цій роботі?

11. Що таке OpenSSL?

12. Які протоколи реалізує OpenSSL?

13. Які криптографічні функції підтримує OpenSSL? (назвати хоча б 3)

14. На яких операційних системах працює OpenSSL?

15. Який алгоритм асиметричного шифрування використовується в лабораторній роботі?

16. Яка команда використовується для генерації приватного ключа в OpenSSL?

17. Яке розширення має файл приватного ключа?

18. Яка команда використовується для генерації публічного ключа в OpenSSL?

19. Яке розширення має файл публічного ключа?

20. Який принцип асиметричного шифрування?

21. Яка команда використовується для підписання документа в OpenSSL?

22. Що означає опція `dgst` в команді підписання?

23. Що таке дайджест?

24. Який алгоритм хешування використовується в прикладі?

25. Що означає опція `-sign` в команді підписання?

26. Яке розширення має файл цифрового підпису?

27. Що відбувається з хешем документа під час підписання?

28. Які файли необхідно передати іншому користувачу для перевірки підпису?

29. Яка команда використовується для перевірки підпису в OpenSSL?

30. Які результати перевірки підпису можливі та що вони означають?

Лабораторна робота № 5. Налаштування брандмауера і VPN

Мета: набуття практичних навичок з налаштування VPN (віртуальної приватної мережі) та брандмауера (firewall) для забезпечення безпечного та захищеного доступу до мережевих ресурсів; ознайомлення з основами налаштування VPN для захищеного з'єднання між віддаленими точками та як правильно конфігурувати брандмауер для контролю за мережевим трафіком і запобігання несанкціонованому доступу.

Обладнання: Персональні комп'ютери з встановленою операційною системою Windows із доступом до Інтернету.

Програмне забезпечення: програма Microsoft Defender, веб-браузер Opera.

Теоретичні відомості

Безпека в операційній системі Windows 10 та 11 являє сама собою захист від вірусів та несанкціонованого доступу. Персональний комп'ютер буде активно захищено з моменту старту ОС Windows. Такий захист постійно перевіряє присутність зловмисних програм (вірусів, шкідливого програмного забезпечення інших загроз). Крім того, автоматично завантажуються оновлення, щоб пристрій залишався в безпеці та був захищений від нових загроз.

Безпека у Windows включає в себе вбудовану антивірусну програму під назвою Антивірус для **Microsoft Defender** (у попередніх версіях Windows 10 цей захист мав назву Захисник Windows Центр безпеки). Якщо ж на відповідному пристрої інстальована й увімкнена інша антивірусна програма, Антивірус для Microsoft Defender автоматично вимикається. У разі

видалення іншої програми, Антивірус для Microsoft Defender знову активується автоматично.

Основні функції та налаштування служби "Безпека у Windows":

1. Захист від вірусів і загроз — здійснюється моніторинг загроз для пристрою, запуск перевірок і отримання оновлень для виявлення нових загроз.

2. Захист облікових записів — доступ до налаштувань для входу в систему та управління обліковим записом, зокрема для **`Windows Hello`** та динамічного блокування.

3. Брандмауер і захист мережі — налаштування брандмауера та моніторинг активності ваших мереж і підключень до Інтернету.

4. Керування програмами та браузерами — налаштування **SmartScreen** для Microsoft Defender, що захищає пристрій від небезпечних програм, файлів, сайтів і завантажень, а також доступ до функції запобігання експлойтам, що дозволяє налаштовувати захист для ваших пристроїв.

5. Безпека пристрою — зміна параметрів безпеки пристрою для захисту від атак зловмисного програмного забезпечення.

6. Продуктивність і справність пристрою — отримання відомостей про стан справності пристрою.

7. Родина — функція для моніторингу дій дітей в Інтернеті та управління пристроями в межах вашої родини або колективу.

Ці інструменти дозволяють забезпечити комплексний захист для пристрою, від вірусів до управління сімейною безпекою в Інтернеті.

Окрім брандмауера важливим елементом роботи з ПК є VPN. **VPN (Virtual Private Network)** — це технологія, яка дозволяє створювати захищене з'єднання між вашим пристроєм

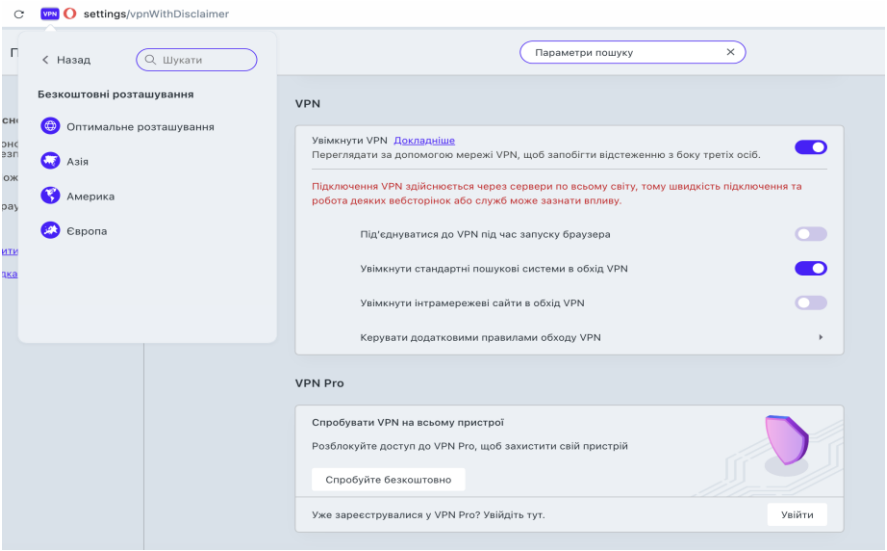
та Інтернетом. Вона забезпечує конфіденційність та безпеку під час перегляду сайтів або використання онлайн-сервісів, приховуючи вашу реальну IP-адресу і замінюючи її адресою сервера VPN. Це дозволяє обійти блокування контенту, доступного лише в певних регіонах, а також приховати вашу географічну локацію, забезпечуючи більшу анонімність.

Крім того, VPN шифрує весь інтернет-трафік, що йде через його сервер, забезпечуючи захист від стеження та зловмисників, що може бути особливо важливо на публічних мережах Wi-Fi. З VPN ви можете бути впевнені, що ваші особисті дані, паролі та банківські транзакції будуть залишатися приватними.

Завдяки VPN можна безпечно підключатися до віддалених мереж, що використовуються компаніями для захисту корпоративної інформації. Це особливо корисно для тих, хто працює з чутливою інформацією або часто перебуває в подорожах. VPN також дозволяє обійти обмеження, накладені на інтернет-ресурси в окремих країнах, забезпечуючи вільний доступ до інформації.

Розглянемо приклад використання VPN у браузері Opera. За замовчуванням, він виключений. Однак, його можна увімкнути. Для цього слід відкрити параметри і переключити перемикач (як показано на рисунку).

Можна вибрати локацію (Азія, Америка, Європа) і додаткові параметри (підєднуватися під час запуску, інтрамережеві сайти, правила обходу VPN). Окрім безкоштовного VPN, можна спробувати Pro, який вимагає реєстрації.



Завдання

1. Відкрийте `Безпека у Windows` для налаштування захисту вашого пристрою: перейдіть у меню `Пуск` → `Настройки` → `Оновлення та захист` → `Безпека у Windows`. У цьому розділі ви побачите піктограми, які вказують на стан безпеки:

- Зелений колір — немає необхідності в діях.

- Жовтий колір — рекомендовано виконати певні дії для покращення безпеки.

- Червоний колір — потребує негайної уваги.

2. Після відкриття `Безпеки у Windows`, виберіть довільних 3 варіанти, що надається у Таблиці 1.

3. Для обраного варіанту налаштуйте два параметри безпеки:

- Зробіть відповідні налаштування.

- Зробіть скріншоти результатів на кожному кроці налаштування.

Варіанти для Таблиці 1:

Номер	Варіант налаштування	Опис налаштування
1	Захист від вірусів і загроз	Налаштуйте перевірки на наявність шкідливих програм та оновлення для нових загроз.
2	Захист облікових записів	Налаштуйте параметри входу (Windows Hello, динамічне блокування).
3	Брандмауер і захист мережі	Налаштуйте параметри брандмауера для моніторингу мережевих з'єднань.
4	Керування програмами та браузерами	Налаштуйте фільтр SmartScreen для захисту від небезпечних файлів, програм і сайтів.
5	Безпека пристрою	Змініть вбудовані параметри безпеки для захисту від атак зловмисного ПЗ.
6	Продуктивність і справність пристрою	Перевірте стан пристрою на наявність проблем із продуктивністю або здоров'ям.
7	Родина	Налаштуйте батьківський контроль для моніторингу активності дітей в Інтернеті та на пристроях.

4. Відкрийте браузер Opera. Налаштуйте VPN, міняючи локації. Порівняйте видимі IP адреси вашого комп’ютера на різних сервісах (https://iplocation.io, https://www.iplocation.net). Результати запишіть у вигляді порівняльної таблиці.

5. Оформіть звіти і захистіть лабораторну роботу.

Контрольні питання

1. Що таке VPN і яка його основна функція?

2. Як VPN забезпечує конфіденційність при використанні Інтернету?

3. Які переваги використання VPN на публічних Wi-Fi мережах?

4. Як VPN може допомогти обійти блокування контенту?

5. Що таке брандмауер і яку роль він виконує в захисті комп'ютера?

6. Як брандмауер допомагає запобігти несанкціонованому доступу до комп'ютера?

7. Як налаштування VPN може змінювати вашу географічну локацію?

8. Як VPN шифрує інтернет-трафік?

9. Яка різниця між стандартним VPN і Pro-версією в браузері Opera?

10. Як можна налаштувати VPN у браузері Opera?

11. Які основні функції включає "Безпека у Windows"?

12. Як працює антивірус Microsoft Defender в операційних системах Windows 10 і 11?

13. Що відбувається, якщо на комп'ютері встановлено інший антивірус замість Microsoft Defender?

14. Які налаштування можна змінити у розділі "Захист від вірусів і загроз"?

15. Як налаштувати функцію "Захист облікових записів" у Windows?

16. Для чого потрібне динамічне блокування в Windows?

17. Як налаштувати брандмауер у Windows для захисту мережі?

18. Які параметри можна налаштувати в функції "Керування програмами та браузерами"?

19. Як фільтр SmartScreen захищає ваш комп'ютер від небезпечних програм?

20. Які налаштування забезпечують безпеку пристрою від атак зловмисного програмного забезпечення?

21. Як перевірити стан продуктивності та справності пристрою в Windows?

22. Для чого потрібна функція "Родина" в Windows?

23. Як налаштувати батьківський контроль в Windows для моніторингу активності дітей в Інтернеті?

24. Які етапи потрібно пройти для налаштування захисту через "Безпеку у Windows"?

25. Як змінюється колір піктограм в "Безпеці у Windows" залежно від стану безпеки?

26. Як використовувати сервіс https://iplocation.io для перевірки IP-адреси?

27. Як можна порівняти різні IP-адреси на сервісах https://iplocation.io та https://www.iplocation.net?

28. Що відбудеться, якщо в операційній системі Windows не увімкнено захист від вірусів і загроз?

29. Як зміна IP-адреси при використанні VPN може допомогти обійти географічні обмеження для доступу до контенту?

30. Як змінюється видимий IP-адрес при зміні локації в браузері Opera за допомогою VPN?

Лабораторна робота № 6. Аналіз фішингових листів і створення алгоритму реагування

Мета: Ознайомитися з ознаками фішингових листів. Навчитися виявляти фішингові листи за допомогою ручного аналізу та автоматизованих інструментів. Розробити алгоритм реагування на фішингові листи.

Обладнання: персональні комп'ютери з встановленою операційною системою Windows із доступом до Інтернету.

Програмне забезпечення: текстовий редактор, антивірусне програмне забезпечення; інструменти для аналізу електронної пошти (наприклад, Email Header Analyzer, онлайн-ресурси для перевірки доменів).

Теоретичні відомості

Фішинг — це вид шахрайства, метою якого є отримання конфіденційної інформації, такої як паролі, дані банківських карток або особисті дані. Основні ознаки фішингових атак включають підроблені URL-адреси, помилки в тексті, термінові заклики до дії та прохання надати особисту інформацію. Зловмисники часто імітують відомі бренди або організації, що допомагає їм викликати довіру у жертв. Фішингові листи зазвичай поширюються через електронну пошту, але також можуть надсилатися через SMS, соцмережі або месенджери. Одним із основних способів розповсюдження фішингових листів є масове розсилання спам-кампаній із підробленими посиланнями. Інший спосіб — зараження комп'ютерів користувачів шкідливим програмним забезпеченням, яке автоматично надсилає листи від імені жертви. Важливим інструментом для виявлення фішингових повідомлень є аналіз

заголовків електронних листів (**email headers**), які містять технічну інформацію про відправника, отримувача та шляхи доставки листа. Дослідження заголовків дозволяє виявити фальшиві адреси відправників або підозрілі сервери, через які були надіслані листи. Аналіз таких полів, як "**Received**", "**From**" і "**Return-Path**", допомагає встановити джерело фішингових повідомлень. Таким чином, знання структури email headers є важливим інструментом у виявленні шахрайських листів і захисті від фішингових атак.

Для виявлення і перевірки повідомлення не предмет фішингу слід

а) перевірити домен відправника MXToolbox, PhishTank або інших ресурсів;

б) проаналізувати посилання, які містить лист наприклад, через VirusTotal, PhishTank або вручну.

в) дослідити заголовок листа (через інструменти типу Mailheader.org) і визначати фальшиві маршрути доставки.

г) оцінити граматику, стиль повідомлення, зокрема помилки у граматиці, нетипові для офіційних організацій.

Типовий приклад фішингового листа наведено нижче.

Відправник: support@amazon..com
Тема: Термінова перевірка замовлення на Amazon
Текст листа:

Шановний користувачу,

Ми виявили підозрілу активність на вашому обліковому записі Amazon. Ваше останнє замовлення не вдалося

підтвердити, тому вам потрібно пройти верифікацію (верифікація).

Для того, щоб уникнути блокування вашого акаунта, будь ласка, підтверджуйте своє замовлення, натиснувши на посилання нижче:

Підтвердити замовлення (https://amason.com/order/hdjcnhdj28djnsd/confirm)

Якщо ви не робили цього замовлення, будь ласка, зв'яжіться з нашою службою підтримки.

З найкращими побажаннями,
Команда Amazon

Як видно з листа, відправник ніби правильний, amazon.com дійсно є такий сайт і домен, однак може виявитися, що ім'я відправника підроблено. Це можна виявити за допомогою аналізу заголовків листа. Досить часто GMAIL автоматично позначає такі листи, як підозрілі. Але якщо хтось користується поштовим клієнтом чи використовує власну пошту, то такий лист пропускається системою. Крім того зловмисники можуть відправляти з некоментованих пошт. Робити разові реєстрації для масової розсилки. Як правило, у посиланнях, які містяться у листі зловмисник лишає ідентифікатор, щоб знати кого мін «впіймав на гачок». Це може бути якісь хеші в url-адресах, незрозумілі параметри. Тому рекомендується зазначені листи (заголовки, текст, відправник) перевіряти різними сервісами,

наприклад Email Header Analyzer (https://mxtoolbox.com/EmailHeaders.aspx).

Завдання

1. Завантажити або скопіювати текст фішингового листа.
2. Визначити ключові підозрілі моменти:

- Помилки у тексті (граматичні, стилістичні).
- Використання термінових повідомлень ("Ваш обліковий запис буде заблоковано").
- Посилання на підозрілі або незнайомі домени.

3. Використати Email Header Analyzer для перевірки заголовків.
4. Сформулюйте послідовність дій для перевірки підозрілого листа (наприклад, перевірка адреси відправника, аналіз посилань, ізоляція підозрілого файлу).
5. Описати дії в разі підтвердження фішингової атаки (повідомлення IT-відділу, блокування домену).
6. Створіть макет фішингового листа (виключно в навчальних цілях) із зазначенням підозрілих елементів.
7. Оформіть звіти і захистіть лабораторну роботу.

Контрольні питання

1. Що таке фішинг?
2. Які основні ознаки фішингових листів?
3. Чому фішингові листи часто використовують термінові заклики до дії?
4. Як підроблені URL-адреси можуть бути використані в фішингових листах?
5. Які найбільш поширені методи фішингових атак?

6. Як зловмисники використовують імітацію відомих брендів для досягнення своїх цілей?

7. Яким чином фішингові листи можуть надсилатися через SMS чи месенджери?

8. Які ресурси можна використовувати для перевірки доменів у фішингових листах?

9. Що таке email header і яку інформацію він містить?

10. Як аналізувати заголовки листів для виявлення фішингу?

11. Які поля заголовків електронних листів важливо перевіряти при аналізі фішингових листів?

12. Як перевірити джерело фішингового листа через поле "Received"?

13. Яку роль у виявленні фішингових листів відіграє поле "Return-Path"?

14. Як оцінка граматики може допомогти виявити фішингові листи?

15. Чому посилання в фішингових листах можуть містити незрозумілі параметри чи хеші?

16. Як перевірити посилання на наявність шкідливих елементів?

17. Що таке VirusTotal і як ним користуватися для перевірки фішингових листів?

18. Які онлайн-ресурси можна використовувати для аналізу доменів у фішингових листах?

19. Як аналізувати текст фішингового листа для визначення підроблених елементів?

20. Як перевірити домен відправника за допомогою MXToolbox?

21. Що таке PhishTank і як цей ресурс допомагає виявляти фішингові сайти?

22. Які інструменти використовуються для перевірки email headers?

23. Як використовувати Email Header Analyzer для аналізу фішингових листів?

24. Чому важливо перевіряти URL-адреси, навіть якщо вони виглядають схожими на офіційні?

25. Як використовувати антивірусне програмне забезпечення для захисту від фішингових атак?

26. Як на практиці перевірити фішингові листи за допомогою аналізу заголовків?

27. Як фішингові атаки можуть використовувати підроблені платіжні сервіси?

28. Як оцінка змісту і структури фішингового листа може допомогти в його розпізнаванні?

29. Чому фішингові листи можуть містити повідомлення про термінові дії або обмеження часу?

30. Як розробити алгоритм реагування на фішингові листи у реальному середовищі?

Частина 2. Воркшопи

Воркшоп 1. Фішинг у дії: аналіз уразливостей та методи захисту

Мета: мета цього воркшопу надати учасникам практичний досвід у виявленні та аналізі фішингових сайтів, розвиваючи навички критичного мислення та уваги до деталей. Учасники зможуть вивчити ознаки підроблених веб-ресурсів, зрозуміти основи їх створення та ознайомитися з ефективними методами захисту від таких загроз. Це допоможе сформувати стійкий підхід до безпечної поведінки в інтернеті та підвищити обізнаність про кіберризики.

Обладнання, програмне забезпечення: персональні комп'ютери для груп; проектор або панель (де демонструється кейс); програма для презентацій (гугл презентації); програма для конференцій (google-meet); розроблені фейкові сайти і локально розміщені на комп'ютерах за допомогою технології контейнеризації.

План:

I. Ознайомлення з кейсом (вступ до теми: огляд фішингових атак, їхні наслідки та реальні приклади).

II. Демонстрація кейсу (показ фейкового сайту на великому екрані або проєкторі; поетапний аналіз: URL, сертифікат безпеки, підозрілі елементи інтерфейсу).

III. Аналіз кейсу в групах (розподіл студентів на групи; кожна група отримує завдання дослідити сайт і знайти ознаки фейковості на основі аналізу елементів сайту: URL, сертифікати, структура сторінки та запис висновків).

IV. Обговорення знахідок (презентація висновків про знайдені ознаки фейковості; обговорення найбільш очевидних ознак фейковості, а які могли залишитися непомітними).

V. Аналіз методів захисту (обговорення кроків, які користувачі можуть вжити, щоб уникнути фішингових сайтів; короткий огляд інструментів і методів, що допомагають у виявленні фішингових сторінок).

VI. Підсумки та ключові уроки (підсумування результатів розбору кейсу; основні рекомендації та практичні поради щодо безпечної поведінки в інтернеті).

Хід роботи

I.

Фішингові атаки стали однією з найпоширеніших кіберзагроз сучасності, спрямованих на обман користувачів для збору їхніх особистих даних. Зазвичай фішинг полягає в імітації реальних сайтів, сервісів або електронних листів, щоб змусити жертву надати конфіденційну інформацію, таку як логіни, паролі, банківські дані тощо. Зловмисники, використовуючи техніки соціальної інженерії, створюють копії сторінок популярних платформ, таких як соціальні мережі, електронні поштові сервіси або онлайн-банкінг, щоб люди не підозрювали обману та довіряли цим фейковим сайтам.

Наслідки таких атак можуть бути дуже серйозними: від крадіжки особистих даних і фінансових втрат до компрометації акаунтів, використаних для подальшого поширення загрози серед контактів жертви. Щороку жертвами фішингу стають тисячі людей та організацій, і хоча багато користувачів знають про існування таких атак, вони все одно піддаються обману через високу схожість фейкових ресурсів із справжніми.

Для глибшого розуміння та обізнаності ми розглянемо кейс, який імітує реальний фішинговий сайт. Це дозволить побачити, як створюються такі обманні ресурси, і зрозуміти, на які ознаки потрібно звертати увагу, щоб уникнути небезпеки. Учасники зможуть дослідити цей сайт, визначити ознаки фальшивості, а також дізнатися, як можна захистити себе від подібних атак у майбутньому.

У рамках цього кейсу буде розіслано студентам декілька електронних листів, які містять посилання на різні веб-сайти. Деякі з цих посилань ведуть на справжні, безпечні ресурси, тоді як інші є фішинговими — вони імітують відомі платформи і мають на меті зібрати особисті дані. Завдання студентів — уважно переглянути ці листи, проаналізувати посилання та виявити фішингові. Використати всі доступні методи перевірки: звернути увагу на URL-адресу, сертифікат безпеки, незвичні елементи на сторінці та інші деталі, що можуть свідчити про підробку. Це допоможе розпізнавати фішингові атаки та зрозуміти, на які аспекти слід звертати увагу в реальних умовах.

II.

На етапі демонстрації кейсу викладач виведе фейковий сайт на великий екран або проектор, щоб усі студенти могли детально ознайомитися з його елементами. Спочатку звернемо увагу на URL-адресу: часто фішингові сайти мають подібний, але не ідентичну адресу справжнього сайту, з помилками або додатковими символами. Далі перевіримо сертифікат безпеки: у більшості фейкових сайтів відсутній SSL-сертифікат, що є сигналом небезпеки, або ж сертифікат ненадійний.

Звернемо увагу на процес реєстрації на фейковому сайті. Зазвичай такі сайти пропонують швидкий вхід через популярні соціальні мережі — Facebook, Google або інші, що створює

ілюзію надійності та зручності для користувача. Ми покажемо, як цей фейковий процес працює: при натисканні на кнопку реєстрації через соцмережу користувача перенаправляють на сторінку, що виглядає як офіційна сторінка авторизації, але насправді є підробленою.

Проаналізуємо URL-адресу такої сторінки, чи є сертифікат безпеки, і як відрізнити оригінальні форми соцмереж від фейкових. Такий аналіз допоможе студентам краще зрозуміти, як фішингові сайти використовують підроблені екрани реєстрації для збору особистих даних, і дасть навички для розпізнавання таких атак у майбутньому.

Перейдемо до візуального аналізу інтерфейсу, розглянемо деталі сторінки, такі як графіка, кнопки, розташування елементів. Часто фішингові сторінки містять незначні помилки або мають спрощений дизайн, через що виглядають менш якісно, ніж оригінал. Також покажемо типові ознаки фішингових форм: поле для введення пароля чи інших конфіденційних даних часто виділене некоректно або запрошує більше даних або менше, ніж зазвичай запитує справжній сайт. Цей покроковий аналіз дозволить студентам навчитися швидко розпізнавати підозрілі елементи на фішингових сайтах.

III.

Ось план, за яким студенти можуть аналізувати кейс:

1. Аналіз URL-адреси.

1.1. Перевірте, чи є в URL-адресі явні помилки або відмінності від оригінального сайту (наприклад, додаткові символи, доменні зміни, ліві домени).

1.2. Подивіться на протокол (HTTP чи HTTPS). Якщо сайт не використовує HTTPS, це сигнал про потенційну небезпеку.

2. Перевірка сертифіката безпеки.

2.1. Клікніть на значок замка в адресному рядку і перевірте сертифікат сайту. Чи є він насправді валідним?

2.2. Перевірте, чи відповідає домен сайту і сертифікат один одному. Якщо сертифікат належить іншому домену або не виданий довіреним центром сертифікації, це ще одна ознака фейкового сайту.

3. Аналіз структури сторінки.

3.1. Перевірте основні елементи сторінки: чи є очевидні помилки в дизайні, нетипові шрифти, неправильно відображені зображення або графіка.

3.2. Оцініть навігацію на сайті. Чи працюють усі посилання? Чи є на сторінці підозрілі чи незрозумілі кнопки/посилання?

4. Перевірка форми реєстрації.

4.1. Зверніть увагу на форму реєстрації чи авторизації. Чи запитує сайт надмірну кількість особистих даних або більше, ніж це зазвичай потрібно на таких платформах?

4.2. Перевірте наявність полів для введення паролів або фінансової інформації (якщо це сайт соцмереж, чому вони потребують ці дані?).

5. Запис висновків.

5.1. Кожна група повинна записати свої спостереження та висновки щодо фейковості сайту. Зокрема, що саме викликає підозри та які елементи були найбільш тривожними.

5.2. Після цього групи повинні обговорити знайдені ознаки фейковості та підготувати коротку презентацію своїх висновків для подальшого обговорення в класі.

IV.

Так, на етапі обговорення знахідок студенти мають створити презентацію, в якій детально опишуть свої висновки та

знайдені ознаки фейковості сайту. Кожна група підготовлює коротку презентацію (5-7 слайдів), де вони:

1. описують загальну структуру сайту, зокрема, виявлені проблеми з дизайном, структуруванням елементів, помилками в графіці чи тексті;
2. вказують на проблеми з URL-адресою: чи є в ній ознаки підробки, чи є помилки, що свідчать про фальшивий ресурс;
3. описують проблеми з сертифікатом безпеки: чи є він валідним, чи відповідає домену, чи не викликає сумнівів;
4. вказують на форми реєстрації: чи запитуються зайві особисті дані, чи є елементи, що відрізняються від стандартних для оригінальних сайтів;
5. кожна група презентує свої висновки щодо найбільш очевидних ознак фейковості сайту (під час презентації студенти можуть навести конкретні приклади та скріншоти із сайту, що підтверджують їхні зауваження);
6. згадують наявність або відсутність непомітні ознаки, які могли бути проігноровані або не відразу привернули увагу (наприклад, незначні помилки в граматиці, неполадки з відображенням кнопок, або незвичні запити до користувача);
7. дають рекомендації з захисту.

V.

Дискусія буде зосереджена на обговоренні можливих методів захисту від фішингових атак. Під час неї студенти повинні поділитися своїми думками про те, як можна захистити себе від фішингових сайтів, використовуючи різні інструменти та практики безпеки. Зокрема, вони обговорюватимуть важливість перевірки URL-адреси, актуальності сертифікатів

безпеки та уважного ставлення до форм реєстрації. Також обговорюватимуть інструменти для перевірки сайту на фішингові загрози, такі як розширення для браузерів, антивірусні програми та додатки для двофакторної аутентифікації. Крім того, буде піднято питання про регулярні оновлення програмного забезпечення та використання антифішингових функцій, що вбудовані в сучасні браузери, а також як важливо навчати користувачів критичному ставленню до запитів на введення особистих даних. Ця дискусія сприятиме формуванню у студентів комплексного підходу до захисту в онлайн-середовищі.

VI.

Підсумовуючи, основні уроки включають важливість уважного перевіряння URL-адрес, сертифікатів безпеки та форм реєстрації, щоб уникнути фішингових атак. Користувачі повинні бути обережними при наданні особистих даних і використовувати інструменти безпеки, такі як двофакторна аутентифікація та антивірусні програми. Основна рекомендація — бути критичними до онлайн-ресурсів і регулярно застосовувати сучасні методи захисту для зниження ризику атак.

Воркшоп 2. Взлом шифру пароля. Як захистити свої паролі від атак

Мета: навчити студентів основним принципам захисту паролів, зокрема створенню безпечних паролів, використанню сучасних інструментів для їх зберігання, а також розумінню методів атак на паролі для кращого запобігання можливим загрозам.

Обладнання, програмне забезпечення: персональні комп'ютери для груп; проектор або панель (де демонструється кейс); програма для презентацій (гугл презентації); програма для конференцій (google-meet); розроблені алгоритми і підготовлені бібліотеки в локальному середовищі для демонстрації взлому.

План

I. Основи паролів та хешування і їхні функції. Що таке пароль і чому його хешують? Принцип роботи MD5. Інші популярні алгоритми хешування (SHA-256, bcrypt).

II. Методи взлому паролів. Брютфорс (перебір). Використання скомпрометованих баз даних. Використання спеціального програмного забезпечення (наприклад, Hashcat). Демонстрація одного з безпечних прикладів (на власному "тренувальному"). Використання заздалегідь згенерованого слабкого пароля. Показ на інструменті hashcat-passords. Порівняння швидкості зламу слабкого пароля та складнішого. Унікальні паролі. Використання алгоритмів хешування. Менеджер паролів.

III. Робота в групах 4-5 груп. Перевірка паролів за допомогою онлайн-сервісів (наприклад, Have I Been Pwned).

Менеджери паролів. Створення презентації (4-5 слайдів) з рекомендаціями щодо паролів.

IV. Доповідь, презентування рекомендацій від кожної групи.

V. Підсумки та ключові уроки (підсумування результатів розбору кейсу; основні рекомендації та практичні поради щодо безпечної поведінки в інтернеті)

Хід роботи

I.

Пароль — це секретна комбінація символів, яка використовується для підтвердження особи користувача та захисту доступу до інформації чи системи. Він може складатися з:

- букв (великих і малих),
- цифр,
- спеціальних символів.

Пароль є ключем до захисту особистих даних, таких як облікові записи в соцмережах, електронна пошта, банківські акаунти тощо.

Зберігати паролі у відкритому вигляді (plain text) є надзвичайно небезпечно. Якщо зловмисники отримають доступ до бази даних, вони одразу зможуть побачити всі паролі. Тому для зберігання паролів часто використовують хешування.

Хешування — це процес перетворення пароля у фіксовану послідовність символів (хеш), яка не може бути безпосередньо перетворена назад у вихідний пароль. Переваги хешування:

Безпека користувачів:

Якщо база даних з хешами зламається, зловмисникам буде значно складніше відновити оригінальні паролі.

Захист від інсайдерів:

Навіть адміністратори системи не можуть побачити справжні паролі користувачів.

Мінімізація шкоди:

У разі витоку хешів зловмисники можуть зламати лише слабкі паролі, але не всі дані одразу.

Ускладнення атак:

Для зворотного обчислення пароля з хешу потрібен значний обсяг ресурсів і часу, особливо якщо алгоритм хешування надійний.

Коли користувач створює пароль система перетворює його в хеш за допомогою спеціального алгоритму (наприклад, MD5, SHA-256, bcrypt). І в базі даних зберігається лише хеш, а не сам пароль. Під час входу користувач вводить пароль, система хешує його і порівнює отриманий хеш із тим, що зберігається у базі. Якщо хеші співпадають, доступ надається. Алгоритми хешування:

```
md2 (довжина 128 біт)
md4 (довжина 128 біт)
md5 (довжина 128 біт)
sha1 (довжина 160 біт)
sha224 (довжина 224 біт)
sha256 (довжина 256 біт)
sha384 (довжина 384 біт)
sha512 (довжина 512 біт)
bcrypt
```

MD2, MD4 та MD5 — це криптографічні хеш-функції, розроблені Рональдом Рівестом у рамках серії Message-Digest

(MD). Вони використовуються для створення унікального "відбитка" (хешу) даних фіксованої довжини. Усі три алгоритми генерують хеш довжиною 128 біт.

SHA (Secure Hash Algorithm) — це сімейство криптографічних хеш-функцій, створених для забезпечення надійного хешування даних. Алгоритми SHA використовуються для перетворення довільних вхідних даних у фіксовану довжину вихідного хешу, що слугує "цифровим відбитком" інформації.

bcrypt — це криптографічний алгоритм для хешування паролів, розроблений спеціально для забезпечення стійкості до атак методом перебору (brute-force). Його створено на основі алгоритму безпечного зберігання паролів. bcrypt включає параметр work factor (або cost), що визначає кількість повторень операцій хешування. Це ускладнює обчислення хешу. bcrypt автоматично додає унікальну сіль до кожного пароля, що запобігає створенню "rainbow tables".

Алгоритм	Довжина хешу	Призначення	Використання
SHA-224	224 біт	Економія пам'яті	Нішеве, менш популярне
SHA-256	256 біт	Загальні потреби	Стандартна криптографія, блокчейн
SHA-384	384 біт	Збалансована стійкість	Цифрові підписи, сертифікати
SHA-512	512 біт	Максимальна безпека	Високозахищені системи, великі дані

II.

Які би паролі не були у зловмисників завжди є план, інструменти, щоб спробувати зламати пароль, доступ чи хеш. Наприклад,

Брутфорс (перебір варіантів). Це метод злому паролів, за якого зловмисник автоматично перебирає всі можливі комбінації символів до знаходження правильного пароля. Цей підхід дуже

повільний і неефективний для складних паролів, але працює для простих чи коротких.

Використання скомпрометованих баз даних. Зловмисники використовують бази даних паролів, отриманих із попередніх витоків інформації, для спроби підбору пароля через їх повторне використання. Цей метод ефективний, якщо користувач застосовує той самий пароль на кількох сервісах.

Використання спеціального програмного забезпечення (наприклад, Hashcat). **Hashcat** — це інструмент для зламу хешів паролів, що підтримує брутфорс, словникові атаки та атаки з використанням комбінованих методів. Він може використовувати графічні процесори (**GPU**) для пришвидшення зламу складних хешів.

На цьому етапі демонструється можливість взлому паролю md5 за до допомогою **hashcat-passords**.

Студентам пропонується придумати прості паролі 4, 5, 3 букв і цифр. Наприклад, «dfi1», «r78f2», «f8k7l0». Ці довільно придумані паролі пропонується захешувати алгоритмом md5. Цей алгоритм популярний і є багато сервісів, що це роблять онлайн. Наприклад, https://www.md5.cz/ Для кожного паролю генеруємо хеш.

«dfi1» — «6bb48c4660dc56ef6cca4c99f2a6a06f»
«r78f2» — «08d4d6ab99f607f825ffde1b5253df27»
«a!2» — «c4ab0c4c9901244047bf6716dcaf8516»

Далі, використовуючи hashcat, демонструємо як на основі хешу можна отримати оригінальний пароль, наприклад

```
┌──(root㉿b338cd19b7af)-[/]
└─# hashcat -m 0 -a 3 "6bb48c4660dc56ef6cca4c99f2a6a06f" ?a?a?a?a
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6,
=============================================================================
* Device #1: cpu--0x000, 1438/2941 MB (512 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Brute-Force
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB

6bb48c4660dc56ef6cca4c99f2a6a06f:dfi1

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 6bb48c4660dc56ef6cca4c99f2a6a06f
Time.Started.....: Tue Dec 17 16:23:32 2024 (0 secs)
Time.Estimated...: Tue Dec 17 16:23:32 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?a?a?a?a [4]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........: 41645.1 kH/s (2.19ms) @ Accel:256 Loops:95 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 19845120/81450625 (24.36%)
Rejected.........: 0/19845120 (0.00%)
Restore.Point....: 207872/857375 (24.25%)
Restore.Sub.#1...: Salt:0 Amplifier:0-95 Iteration:0-95
Candidate.Engine.: Device Generator
Candidates.#1....: s98j ->  %#t

Started: Tue Dec 17 16:23:30 2024
Stopped: Tue Dec 17 16:23:34 2024
```

Зі скрину видно, що відповідний пароль було взломано на 4 секунди

```
6bb48c4660dc56ef6cca4c99f2a6a06f:   dfi1
Started: Tue Dec 17 16:23:30 2024
Stopped: Tue Dec 17 16:23:34 2024
```

Аналогічно робимо для двох інших

```
08d4d6ab99f607f825ffde1b5253df27:   r78f2
Started: Tue Dec 17 16:27:39 2024
Stopped: Tue Dec 17 16:28:06 2024
```

і

```
c4ab0c4c9901244047bf6716dcaf8516:   a!2
Started: Tue Dec 17 16:41:12 2024
Stopped: Tue Dec 17 16:41:15 2024
```

Як видно для взлому пароля довжиною 4 символи витрачено 4 секунди, а для 5 символів — 26 с. Хеш пароль довжиною 3 символів зламано за 3 секунди. Якщо ж би задати точнішу маску, то злам відбудеться ще скоріше.

Порівняємо алгоритм sha128 або sha1 (https://emn178.github.io/online-tools/sha1.html). Закодуємо фразу:

«a!2» – «753ed47d999c90ebcd20bd928244397485946ca5»

То для її взлому витрачено 22+53 = 75 секунд:

```
753ed47d999c90ebcd20bd928244397485946ca5:  a!2
Started: Tue Dec 17 16:50:38 2024
Stopped: Tue Dec 17 16:50:53 2024
```

Звідси видно, що sh1 краще від md5. Однак ми-користувачі не можемо керувати алгоритмами, які задіяні для хешування паролів, але можемо забезпечити надійність шляхом генерації довгого, надійного паролю.

Приклади поганих паролів

123456, qwert, 123; 111; qwerty; qazwsx; qazwsxedc; password; «ваш логін»; «номер телефону»;

«дата народження» і т.д.

Поганий пароль	Добрий пароль	Ефективний пароль
`123456`	`Password123!`	`A3r9!0vE2*`
`qwerty`	`Iloveyou!2024`	`Z7&v*eK0$Q2i`
`password`	`Sunshine@2024`	`T1g@!Lq4$9X`
`admin123`	`SecureP@ssw0rd`	`Qw3rTy!#12@!pU1Z`
`letmein`	`Welcome2024!`	`Gr3@t#P@ssw0rD$5`

У таблиці подано приклади поганих, добрих і ефективних паролів.

Запам'ятовування ефективних паролів може бути складним завданням, особливо якщо вони є довгими та складними, тому тут слід використовувати менеджер паролів. **Менеджери паролів** — це програми або онлайн-сервіси, що зберігають паролі в зашифрованому вигляді. Вони:

1. Дозволяють створювати складні паролі для кожного акаунта і запам'ятовувати лише один головний пароль.
2. Зберігають всі паролі в одному місці.
3. Автоматично заповнюють поля для входу на сайти.
4. Генерують складні паролі для вас.

5. Всі паролі зашифровані, і доступ до них має лише головний пароль.

Розглянемо менеджер паролів KeePassXC https://keepassxc.org/, який годиться під усі платформи. На цьому етапі демонструється менеджер паролів KeePassXC, як його додати, встановити і як користуватися.

III.

На цьому етапі студентів ділять на групи, де вони мають змогу самостійно згенерувати пароль, користуватися менеджером паролів. Також пропонується сервіс «Pwned Passwords» https://haveibeenpwned.com/Passwords, де можна перевірити чи придуманий пароль не є скомпроментований і чи не надто простий. На основі прослуханого, демонстрації і симуляції кожна група підготовлює коротку презентацію (5-7 слайдів), де вони:

1. описують загальні проблеми безпеки паролі і хешів;
2. вказують на проблеми з простими паролями, і зі способами їхнього зберігання;
3. пропонують свої додаткові рекомендації щодо захисту паролів, на приклад багатофакторна автентифікація;
4. кожна група презентує свої висновки (під час презентації студенти можуть навести конкретні приклади , що підтверджують їхні зауваження);
5. дають рекомендації з захисту.

IV.

Під час демонстрації презентацій доповідей дискусія буде зосереджена на обговоренні можливих методів захисту, створення паролів. Під час неї студенти повинні поділитися своїми думками про те, як можна захистити себе від взлому. Крім того, буде піднято питання про регулярні оновлення паролів, забезпечення їхньої конфіденційності. Ця дискусія сприятиме формуванню у студентів комплексного підходу до захисту в онлайн-середовищі.

V.

Підсумовуючи, наводимо конкретні висновки, які зроблені самими студентами, узагальнюємо матеріал, коротко робимо підсумок проведеного воркшопу. Основна рекомендація — бути критичними паролів і регулярно застосовувати сучасні методи захисту для зниження ризику атак.

Воркшоп 3. Проведення симуляції мережевої атаки та дослідження відкритих портів на серверах

Мета: надати учасникам базові знання про мережеві атаки, способи їх виявлення та методи захисту. Учасники навчаться аналізувати вразливості серверів і розробляти рекомендації для забезпечення безпеки в мережі.

Обладнання, програмне забезпечення: персональні комп'ютери для груп; проектор або панель (де демонструється кейс); програма для презентацій (гугл презентації); програма для конференцій (google-meet); розроблені алгоритми і підготовлені бібліотеки в локальному середовищі для демонстрації Ddos-атаки. Програми для аналізу логів персонального комп'ютера

План:

I. Вступ.

II. Мережеві атаки на сервери на персональні комп'ютери. Ddos-атака на віддалені сервери. Визначення відкритих портів і можливих вразливостей. Аналіз логів.

3. Індивідуальна робота. Аналіз ір-адресів відомих ресурсів. Аналіз їхньої вразливості. Підготовка презентацій щодо захисту у мережі.

4. Доповідь, презентування рекомендацій.

5. Підсумки та ключові уроки (підсумування результатів; основні рекомендації та практичні поради щодо безпечної поведінки в інтернеті).

Хід роботи

I.

У сучасному світі інформаційних технологій мережеві атаки стали однією з найсерйозніших загроз для безпеки даних та систем. Кожного дня організації та звичайні користувачі стикаються з ризиками, пов'язаними з витоком конфіденційної інформації, зламом систем або блокуванням доступу до ресурсів через різноманітні типи атак.

Мета воркшопу — надати базові знання про те, як розпізнавати мережеві атаки, аналізувати їх сліди у системі, а також розробляти ефективні рекомендації для захисту.

У рамках заходу буде:

- проведена демонстрація ddos-атаки на сервер;
- розглянуто способи аналізу відкритих портів на серверах;
- запропоновано як захищатися у мережі та будувати стратегію безпечної поведінки.

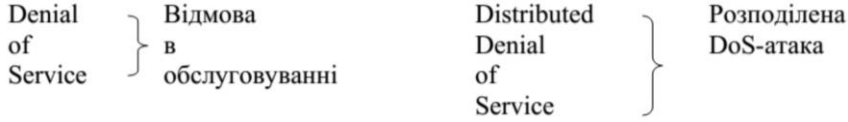
Практична частина воркшопу дозволить кожному учаснику спробувати себе в ролі аналітика безпеки, виявляючи потенційні загрози та розробляючи ефективні заходи для їх запобігання.

Учасники отримають змогу працювати як індивідуально, так і в групах, щоб навчитись вирішувати реальні кібербезпекові виклики та презентувати свої рішення. Безпека в інтернеті починається з розуміння загроз і способів їх уникнення.

II.

Відмова в обслуговуванні (**DoS** або **DDoS**-атаки) — це дії, спрямовані на порушення доступу до ресурсу. Якщо атака виконується лише з однієї IP-адреси, це DoS-атака; якщо використовується багато джерел — DDoS. Основна мета DDoS-

атак полягає в тому, щоб перевантажити сервер великою кількістю запитів через відкритий або дозволений порт, перевищуючи його можливості обробки. У результаті сервер перестає відповідати на запити, і легітимні користувачі не можуть отримати доступ до ресурсу. Це приклад порушення принципу доступності інформації.



Мета DDoS-атаки:

1. Порушення доступності ресурсу (заблокувати доступ до сайту, сервісу або мережі для законних користувачів).

2. Фінансові збитки (викликати простої в роботі компанії, що призводить до втрати прибутку).

3. Дискредитація (знизити репутацію організації, демонструючи її неспроможність захищати власні ресурси).

4. Відволікання уваги (створити умови для інших атак, поки організація фокусується на усуненні наслідків DDoS).

5. Шантаж (вимагати викуп за припинення атаки).

Як боротися з DDoS-атаками:

1. Використання систем захисту від DDoS (інтегрувати спеціалізовані рішення, такі як Cloudflare, Akamai або інші сервіси, які автоматично виявляють і блокують підозрілий трафік).

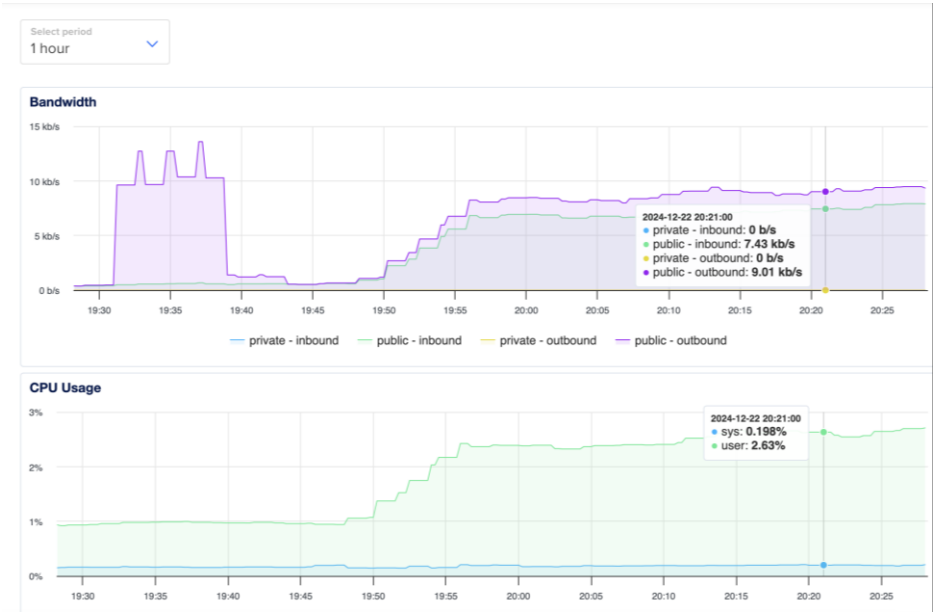
2. Налаштування фільтрації трафіку (застосувати брандмауери, чорні списки IP-адрес або геоблокування для відсіювання небажаних запитів).

3. Збільшення пропускної здатності (використовувати масштабовану інфраструктуру, яка здатна витримати велику кількість запитів, наприклад, за допомогою CDN (мережі доставки контенту)).

4. Моніторинг трафіку в реальному часі (постійно відстежувати активність у мережі, щоб швидко виявляти і реагувати на підозрілі сплески навантаження).

5. Резервні плани та стратегії (мати чіткий план дій на випадок атаки, включаючи перемикання на резервні сервери або використання альтернативних каналів зв'язку).

Розглянемо приклад Dos атаки на наш тестовий веб-сервер. Веб-сервер розміщено на хостингу DiginalOcean, який надає можливість моніторити навантаження. На даний момент, як видно зі скрину, навантаження майже відсутнє.



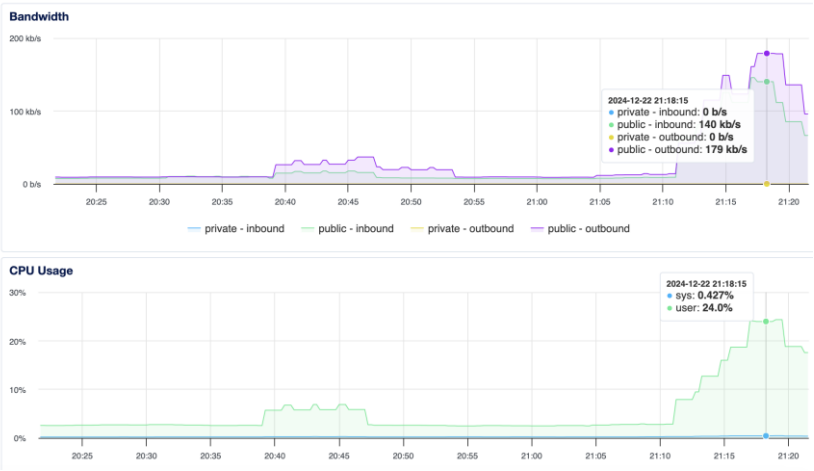
Зімітуємо атаку на нього засобами alpine/bombardier системи, яка вбудована у докер-контейнер. Ми задамо одночасних 200 з'єднань тривалістю 10 секунд кожна і виведемо статистику.

```
docker run -ti --rm alpine/bombardier -c 200 -d 10s -l http://165.22.88.174/
```

Як видно зі статистики сервер справився з такою кількістю

```
olhaleshko@MacBook-Air-Olha ~ % docker run -ti --rm alpine/bombardier -c 200 -d 10s -l http://165.22.88.174/
Bombarding http://165.22.88.174:80/ for 10s using 200 connection(s)
[=========================================================================================================================]
Done!
Statistics        Avg      Stdev        Max
  Reqs/sec        11.89      28.48     158.85
  Latency         6.83s      4.41s     10.27s
  Latency Distribution
     50%     10.18s
     75%     10.21s
     90%     10.23s
     95%     10.23s
     99%     10.25s
  HTTP codes:
    1xx - 0, 2xx - 117, 3xx - 0, 4xx - 0, 5xx - 0
    others - 200
  Errors:
       timeout - 200
  Throughput:    71.47KB/s
olhaleshko@MacBook-Air-Olha ~ %
```

Якщо з збільшити кількість з'єднань до 2000, то на графіку навантажень відразу помітно буде, що навантаження зросло до 24%.



Якщо ж кількість з’єднань збільшити ще на порядок, то з великою ймовірністю сервер перестане працювати.

Аналізуючи навантаження на процесор і пропускну спроможність сервера можна бачити нетипові значення. Саме такі навантаження можуть бути індикатором атаки на сервер.

Вище було зазначено, які заходи слід вжити для запобігання і переривання атак.

Використовуючи різні сервери, для розміщення своїх даних, ресурсів, можна перевірити, які порти відкриті. Це дасть змогу проаналізувати можливі вразливості. Для цього пропонуємо програму **Nmap**.

Nmap (Network Mapper) — це широко використовуваний інструмент для сканування мереж і перевірки безпеки. Він дозволяє виявляти активні хости в мережі, визначати відкриті порти та ідентифікувати служби, що працюють на цих портах, а також виявляти вразливості та конфігураційні проблеми в мережевих пристроях. Основні можливості Nmap:

1. Сканування відкритих портів: Nmap дозволяє перевіряти, які порти відкриті на віддаленому хості або сервері; порти можуть бути фільтровані або закриті брандмауерами, і Nmap дає можливість визначити їх статус.

2. Визначення служб і версій: після виявлення відкритих портів Nmap може визначити, які саме служби працюють на цих портах, і навіть виявляти версії програмного забезпечення, що їх обслуговує; для цього використовуються скрипти (сценарії) або збирання даних через зв'язок з службами, наприклад, HTTP, SSH, FTP тощо.

3. Сканування за допомогою скриптів (Nmap Scripting Engine): Nmap містить велику бібліотеку скриптів, які дозволяють автоматично перевіряти мережі на наявність вразливостей, таких як SQL ін'єкції, проблеми з безпекою веб-додатків, тощо; скрипти можна використовувати для виявлення відомих вразливостей, перевірки конфігурацій безпеки та іншого.

4. Визначення операційної системи: Nmap може спробувати визначити операційну систему, яка працює на віддаленому хості, а також її версію та інші характеристики (наприклад, тип брандмауера).

5. Пошук вразливостей: Nmap містить сценарії для виявлення відомих вразливостей у різних мережевих службах і пристроях. Це дозволяє швидко і ефективно перевірити наявність критичних вразливостей у мережі.

7. Аналіз топології мережі: Nmap дозволяє отримувати уявлення про структуру мережі, визначати активні хости та маршрути між ними.

Наприклад,

Виявлення операційної системи:

nmap -O 165.22.88.174

```
    Starting Nmap 7.95 ( https://nmap.org ) at
2024-12-22 21:43 EET
    Nmap scan report for 165.22.88.174
    Host is up (0.043s latency).
    Not shown: 995 closed tcp ports (reset)
    PORT     STATE    SERVICE
    22/tcp   open     ssh
    25/tcp   filtered smtp
    80/tcp   open     http
    3306/tcp open     mysql
    8080/tcp open     http-proxy
    Device type: general purpose|router
    Running: Linux 5.X, MikroTik RouterOS 7.X
    OS    CPE:    cpe:/o:linux:linux_kernel:5
cpe:/o:mikrotik:routeros:7
cpe:/o:linux:linux_kernel:5.6.3
    OS details: Linux 5.0 - 5.14, MikroTik
RouterOS 7.2 - 7.5 (Linux 5.6.3)
    Network Distance: 9 hops

    OS detection performed. Please report any
incorrect results at https://nmap.org/submit/
.
    Nmap done: 1 IP address (1 host up) scanned
in 3.39 seconds
```

Сканування всіх портів та визначення версій служб:

nmap -sV 165.22.88.174

```
    Starting Nmap 7.95 ( https://nmap.org ) at
2024-12-22 21:45 EET
    Nmap scan report for 165.22.88.174
    Host is up (0.046s latency).
    Not shown: 995 closed tcp ports (reset)
    PORT     STATE    SERVICE VERSION
    22/tcp    open       ssh      OpenSSH 9.7p1
Ubuntu 7ubuntu4 (Ubuntu Linux; protocol 2.0)
    25/tcp   filtered smtp
    80/tcp   open     http    nginx 1.21.6
    3306/tcp open     mysql   MySQL 5.7.34
    8080/tcp  open       http      Apache httpd
2.4.38 ((Debian))
    Service    Info:    OS:    Linux;    CPE:
cpe:/o:linux:linux_kernel

    Service detection performed. Please report
any        incorrect        results        at
https://nmap.org/submit/ .
    Nmap done: 1 IP address (1 host up) scanned
in 10.56 seconds
```

Сканування виявило відкриті порти для SSH, nginx, MySQL і Apache, що може бути вразливим, якщо не застосовано належне обмеження доступу та захист. Необхідно регулярно оновлювати програмне забезпечення та забезпечити захист від несанкціонованого доступу.

Щодо Nmap, то він є безкоштовним і з відкритим кодом, що дозволяє розширювати його функціональність. Nmap є одним з найбільш потужних інструментів для аналізу мереж, і завдяки

своїм можливостям він є незамінним у проведенні тестування на проникнення та оцінки безпеки мереж.

Його можна встановити на усі платформи:

1) MAC. У терміналі виконати команду:

```
brew install nmap
```

2) Ubuntu/Linux. У терміналі виконати команду:

```
sudo apt update
sudo apt install nmap
```

3) Windows:

Завантажити інсталятор з офіційного сайту `Nmap`.

Встановити програму, слідуючи інструкціям.

Запустити через `cmd` або PowerShell.

III.

На цьому етапі студенти працюють самостійно. Студентам пропонується розглянути роботу консольної утіліти Nmap. Пропонується дослідити ір-адреси відомих веб-сайтів, ресурсів на предмет відкрити портів, наявності різних версій програмного забезпечення. Отримані результати пропонується проаналізувати за допомогою матеріалів конспектів лекцій, довідки програми та за допомогою штучного інтелекту (GPT / Gemini).

Отримані результати слід оформити у презентації доповіді на 4-5 слайдів. Там же слід навести пропозиції щодо захисту ресурсів на серверах.

IV.

Під час демонстрації презентацій доповідей дискусія буде зосереджена на обговоренні можливих методів захисту і

відкритих портах. Під час неї студенти повинні поділитися своїми думками про те, як можна захистити себе від взлому і атак. Ця дискусія сприятиме формуванню у студентів комплексного підходу до захисту в онлайн-середовищі.

V.

Під час воркшопу учасники ознайомились з двома типами мережевих атак та методами захисту від них. Були продемонстровані практичні аспекти, як визначити вразливості через сканування мережі за допомогою інструментів, таких як Nmap, а також способи виявлення відкритих портів та небезпечних сервісів на віддалених хостах.

Особливу увагу було приділено основним атакам на локальні мережі і сервери, зокрема DDoS, їхнім можливим наслідкам для безпеки. Учасники мали змогу попрактикуватися в аналізі мережевого трафіку для виявлення ознак атак. Крім того, вони розглянули найкращі практики щодо захисту інформаційних систем, а також дізналися, як налаштовувати основні засоби безпеки, такі як брандмауери та шифрування з'єднань.

У рамках практичної частини учасники працювали з реальними сценаріями, вивчаючи та презентуючи рекомендації для забезпечення безпеки в мережі. Підсумки воркшопу допомогли підкреслити важливість постійного моніторингу мережі та систем, а також необхідність актуалізації програмного забезпечення для запобігання можливим атакам.

Воркшоп 4. Перехоплення та аналіз мережевого трафіку. Шифрування інформації

Мета: вивчення методів перехоплення та аналізу мережевого трафіку, засобів шифрування інформації та практичного використання сучасних інструментів для захисту даних. Учасники отримають навички виявлення вразливостей, захисту даних у мережах і шифрування файлів та повідомлень.

Обладнання, програмне забезпечення: персональні комп'ютери для груп; проектор або панель (де демонструється кейс); програма для презентацій (гугл презентації); програма для конференцій (google-meet); розроблені алгоритми і підготовлені бібліотеки в локальному середовищі для демонстрації перехоплення повідомлень.

План:

I. Вступ.

II. Демонстрація перехоплення повідомлень. Шифровані і не шифровані повідомлення. Методи шифрування. Захищені і не захищені точки wi-fi.

III. Практична частина. Індивідуальна робота. Перевірка, месенджерів, сертифікатів безпеки сайтів. Налаштування шифрування файлів. Самостійне шифрування повідомлень шифром Цезаря. Підготовка презентацій щодо захисту повідомлень.

IV. Доповідь, презентування рекомендацій від кожної.

V. Підсумки та ключові уроки (підсумування результатів розбору кейсу; основні рекомендації та практичні поради щодо безпечної поведінки в інтернеті).

Хід роботи

I.

На цьому воркшопі буде розглянуто інструменти для перехоплення та аналізу мережевого трафіку. У сучасному світі, де інформація передається переважно через мережі, розуміння способів захоплення і захисту даних є важливою складовою кібербезпеки. Ми розглянемо три ключові сценарії перехоплення: у локальній мережі, через атаки "людина посередині" (MITM), а також у незахищених Wi-Fi точках доступу.

Під час демонстрацій учасники дізнаються, як зловмисники можуть перехоплювати пакети, аналізувати їх зміст і розрізняти зашифрований та незашифрований трафік. Особливу увагу приділимо етичним аспектам, включаючи законність та відповідальність при використанні таких інструментів. Також ми розглянемо основні методи захисту від атак, зокрема HTTPS, VPN та засоби запобігання. Цей воркшоп допоможе краще зрозуміти вразливості мереж, а також надасть навички для захисту своїх даних у реальному світі.

II.

Перехоплення повідомлень у локальній мережі — це метод аналізу мережевого трафіку, який використовується для діагностики, моніторингу або забезпечення безпеки. Одним із популярних підходів є пасивне перехоплення, коли дані перехоплюються без зміни їх вмісту або порушення роботи мережі. Однак навіть пасивне перехоплення веде до того, що паролі, персональні дані можуть бути скомпрометовані. Для захисту від таких дій пропонується

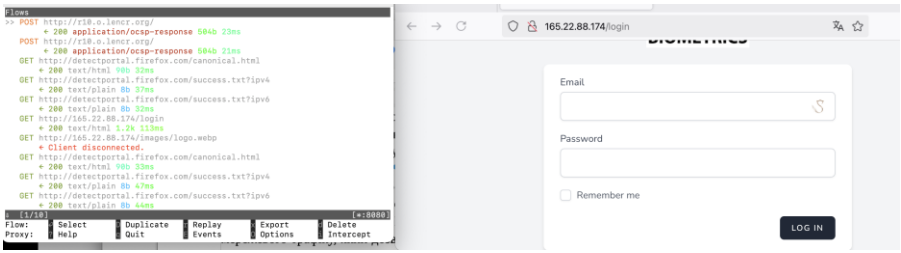
- Використовуйте шифрування трафіку (наприклад, HTTPS).
- Встановіть фільтрацію MAC-адрес.
- Впровадьте моніторинг активності у мережі.

Іншим варіантом потрапити «на гачок» є незахищені Wi-fi мережі. Досить часто пристій користувача може автоматично під’єднатися до доступної безкоштовної Wi-Fi точки. В таких мережах повідомлення, що надсилаються через http-трафіком можуть бути легко перехоплені і прочитані. Тому ніколи не слід підключатися до незахищених Wi-Fi мереж, якщо не використовуєте VPN.

Ще одним варіантом перехоплення повідомлень є атаки "Людина посередині" (MITM). Зараз розглянемо яким чином можна перехопити логін і пароль. Для демонстрації використаємо браузер Firefox. Паралельно запустимо Mitmproxy – це інструмент для аналізу мережевого трафіку, який дозволяє перехоплювати, переглядати та змінювати HTTP/HTTPS-трафік у реальному часі. Він може

1. Перехоплювати трафік. Працює як проксі-сервер для HTTP/HTTPS і WebSocket.
2. Аналізувати дані. Відображає детальну інформацію про запити та відповіді.
3. Змінювати трафік. Можна змінювати дані на льоту.

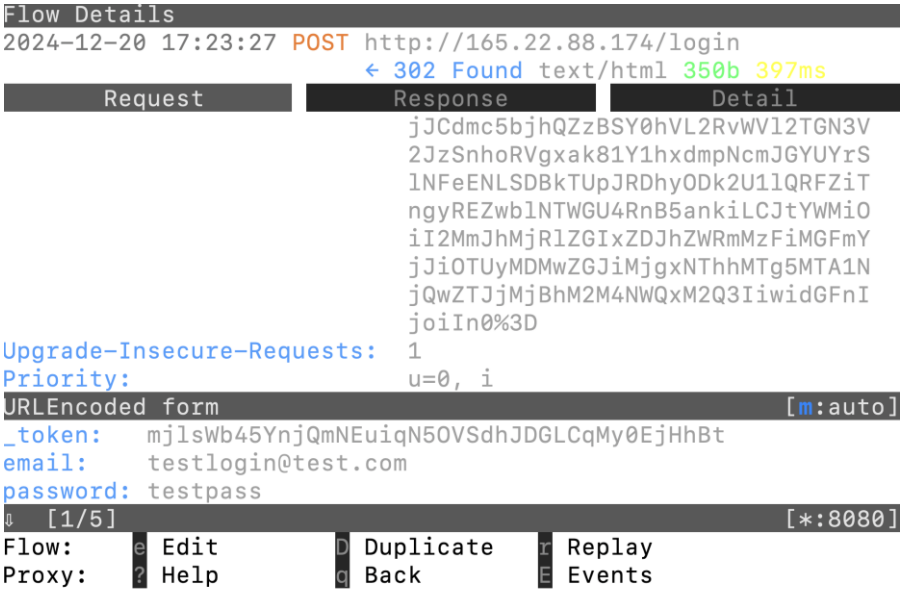
На цьому етапі викладач запускає Mitmproxy через докер і налаштовує проксі у Firefox.



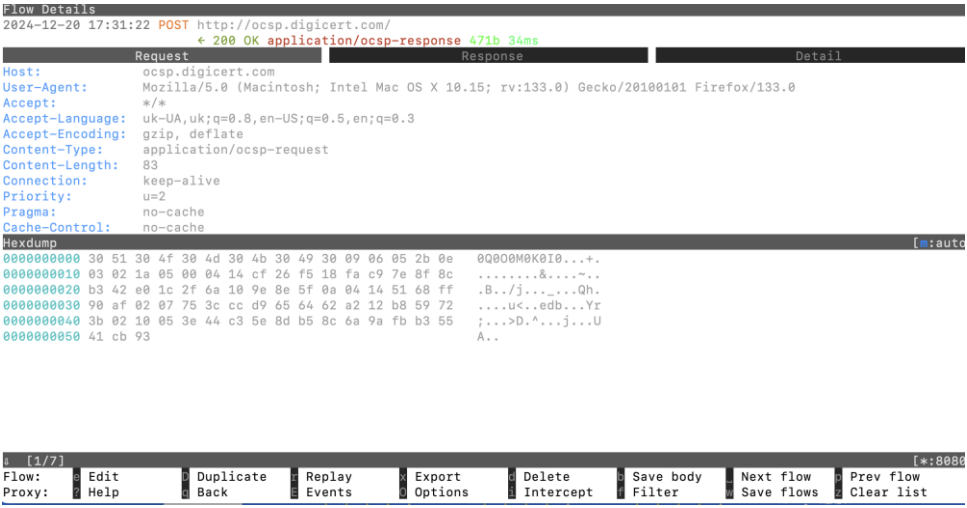
На скрині видно, що відкривши сайт, ми бачимо усі запити, що йдуть з сайту на сервер. Спробуємо ввести логін і пароль

На перехопленні ми бачимо запит login

Якщо відкрити детальніше, то можна безпосередньо побачити, що було введено у поля логін і пароль. Це видно на скрині



Зі скрину видно як логін так і пароль, що вводилися.



Як видно, це так дані легко перехоплюються, якщо відкрити http-сайт. Коли ж відкрити безпечний https, то паролів не видно

Це пояснюється тим, що дані шифруються і передаються через мережу. І просте перехоплення нічого не дає.

Аналогічно можна перехопити вміст файлу та інші дані. Тому рекомендується дуже важливу інформацію шифрувати. Для цього є таке поняття як криптографія.

З грецької «криптографія» перекладається як «тайнопис». Базове значення криптографії полягає в утаємничуванні потрібної інформації. Сучасна криптографія дає інструменти для захисту інформації і у зв'язку з цим є складовою діяльності щодо убезпечення інформації.

Є різні методи втаємничення інформації:

- приховування джерела й каналу передачі повідомлення;
- маскування самого змісту повідомлення;
- ускладнення самої можливості перехоплення повідомлення зловмисником;
- Інші.

Слід відзначити, що криптографія не «приховує» повідомлення, а змінює його форм так, що вона стає недоступною для розуміння зловмисником. Таке перетворення відбувається з використанням відповідних криптографічних систем.

Криптографічними перетвореннями є:

Шифрування – процес перетворення вихідного повідомлення в зашифроване за допомогою шифруючої функції з секретним ключем шифрування відповідно до алгоритму шифрування.

Розшифрування – зворотній процес до шифрування, що полягає в перетворенні зашифрованого повідомлення у вихідне за допомогою функції розшифрування з тим самим секретним ключем відповідно до алгоритму шифрування.

Сукупність перетворень шифрування і розшифрування зветься шифром. Загалом криптосистема має вигляд:

Один з найстаріших і найпростіших видів шифрування, — шифр зсуву (Цезаря), який працює за принципом зсуву кожної літери в алфавіті на певну кількість позицій. Відомий також як шифр Цезаря, він був використаний Юлієм Цезарем для захисту своїх повідомлень.

Принцип роботи шифру зсуву:

1. Кожна буква в тексті зсувається на певну кількість позицій в алфавіті.

2. Наприклад, для зсуву на 3:

- А стає D
- Б стає Г
- В стає Д
- І так далі.

Якщо літері в алфавіті немає місця для зсуву (наприклад, Z у латинському алфавіті), то вона повертається на початок алфавіту:

- Z стає C.

Наприклад:

Шифрування

1. Оригінальний текст: "Секрет"

2. Вибір зсуву: вибираємо зсув на 3 (тобто кожну букву зсуваємо на три позиції вперед).

3. Зашифрований текст:

- С → Ф
- Е → Ж
- К → Н
- Р → У
- Е → Ж
- Т → Х

Отже, шифрований текст буде: "Фжнужх"

Дешифрування

Щоб розшифрувати текст, потрібно застосувати зворотний зсув, тобто зсуваємо букви на кількість позицій у зворотному напрямку.

- Ф → С
- Ж → Е
- Н → К
- У → Р
- Ж → Е
- Х → Т

Зашифрований текст "Фжнужх" повертається до оригінального тексту "Секрет".

З одного боку такий шифр легкий, швидко шифрується та дешифрується. Однак такий шифр легко піддається атаці методом підбору, оскільки кількість можливих зсувів обмежена (для латинського алфавіту це 26 варіантів, для кирилиці — 33).

Для шифрування фалів пропонуємо **Cryptomator** (https://cryptomator.org/downloads/) — це зручний інструмент для шифрування файлів перед завантаженням їх у хмару, наприклад, на Google Drive, Dropbox або інші сервіси. Програма є криптоплатформа, тому годиться для Windows, macOS, Linux, Android або iOS.

1. Запустіть Cryptomator після завершення інсталяції.
2. Натисніть на "Create New Vault" (Створити нове сховище).
3. Виберіть місце для зберігання зашифрованого сховища. Це може бути ваша локальна папка або папка на хмарному сервісі, який ви синхронізуєте (наприклад, Google Drive або Dropbox).
4. Введіть ім'я для вашого сховища.
5. Встановіть пароль для шифрування. Це дуже важливий етап, оскільки цей пароль використовується для доступу до зашифрованих файлів. Переконайтеся, що використовуєте надійний пароль.
6. Після створення сховища, ви отримаєте віртуальний диск, на який можна додавати файли.

Додавання файлів у зашифровану теку:

1. Відкрийте Cryptomator і виберіть ваше створене сховище.
2. Введіть пароль для доступу.
3. Після відкриття сховища ви побачите зашифровану папку (наприклад, "Vault").

4. Перетягніть файли, які потрібно зашифрувати, у цю папку. Файли будуть автоматично зашифровані.

Завантаження файлів в хмару (необов'язково):

1. Після додавання файлів до сховища просто синхронізуйте папку з хмарним сервісом (Google Drive, Dropbox, або іншим), як зазвичай.
2. Файли будуть зашифровані перед завантаженням, тому навіть якщо хтось отримує доступ до вашої хмари, вони не зможуть побачити вміст файлів без пароля.

Доступ до зашифрованих файлів:

1. Щоб відкрити зашифровані файли, знову запустіть Cryptomator і виберіть відповідне сховище.
2. Введіть пароль.
3. Після цього ви зможете переглядати, редагувати або копіювати файли, як з звичайного віртуального диска.
4. Коли ви завершите роботу з файлами, обов'язково закрийте сховище через Cryptomator, щоб зберегти їх у зашифрованому вигляді.

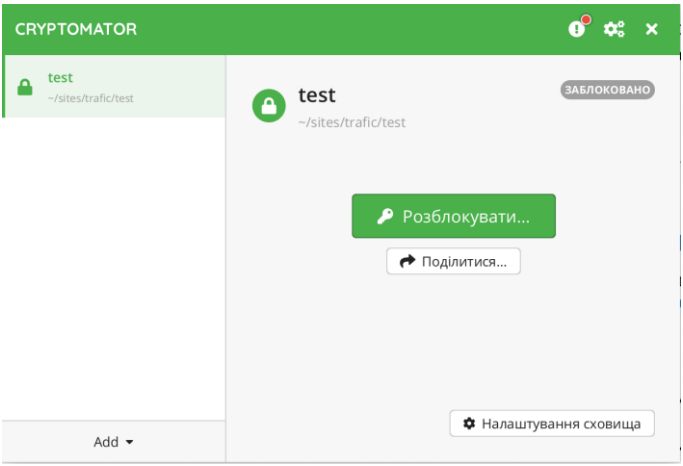
Закриття сховища:

1. Після завершення роботи з файлами, не забудьте закрити сховище через Cryptomator.
2. Це важливо для збереження шифрування файлів. Ви можете бути впевнені, що ваші дані захищені навіть коли файли знаходяться в хмарі.

Важливі зауваження:

1. Якщо ви забудете пароль, доступ до ваших файлів буде неможливим, тому зберігайте його в надійному місці.

2. Переконайтеся, що ваші хмарні сервіси синхронізуються автоматично, щоб зміни в зашифрованих файлах відображалися без проблем.



Cryptomator є простим у використанні інструментом для шифрування файлів перед їх зберіганням у хмарі, що дозволяє захищати конфіденційність даних у будь-який час.

III.

На цьому етапі студенти працюють самостійно. Їм пропонується познайомитися з **Cryptomator**, який попередньо встановлений на комп'ютерах. Студенти додають файли і папки до створених сховищ, перевіряють меню програми, експериментують з налаштуванням.

Другим завданням є створення шифру зсуву (шифру Цезаря). Тут студенти шифрують повідомлення і передають один одному, вказуючи ключ зміщення. Також студенти отримують від викладача перелік зашифрованих повідомлень. Таким чином

пробують розшифрувати повідомлення. Також пропонується ознайомитися з онлайн сервісами, що шифрують інформацію, наприклад https://hostciti.net/calc/it/.

Наступному етапі готуються короткі презентації 3-4 слайди, де вказується важливість безпеки повідомлень і шифрування даних.

На основі прослуханого, демонстрації, виконаних завдань заслуховується короткі висновки бажаючих щодо передачі повідомлень і шифрування даних.

IV.

Під час демонстрації презентацій доповідей дискусія буде зосереджена на обговоренні можливих методів захисту і шифруванні даних. Під час неї студенти повинні поділитися своїми думками про те, як можна захистити себе від взлому і перехоплення даних. Ця дискусія сприятиме формуванню у студентів комплексного підходу до захисту в онлайн-середовищі.

V.

Підсумовуючи, наводимо конкретні висновки, які зроблені самими студентами, узагальнюємо матеріал, коротко робимо підсумок проведеного воркшопу. Основна рекомендація — бути критичними та уважними і регулярно застосовувати сучасні методи захисту для зниження ризику атак.

Частина 3. Індивідуальні завдання

Індивідуальне завдання 1. Аналіз реальних кейсів кіберзагроз

Мета завдання: вивчити та проаналізувати реальні приклади кіберзагроз, визначити типи атак, наслідки для організацій та запропонувати заходи для захисту.

Обґрунтування завдання: аналіз реальних кейсів кіберзагроз в організаціях дозволяє не лише вивчити методи атак і їхні наслідки, але й дає можливість визначити ключові слабкості в існуючих системах захисту. Вивчення таких інцидентів допомагає розробити ефективні стратегії для запобігання майбутнім загрозам і вдосконалення політик безпеки. Крім того, цей аналіз сприяє підвищенню обізнаності і створенню культури безпеки в організаціях, що критично важливо в умовах сучасних кіберзагроз.

Завдання

1. Оберіть один реальний кейс кіберзагрози, що стався в останні кілька років (наприклад, атака на компанії, державні установи, хакерські групи тощо). Відповідний кейс знайдіть в інтернеті.

Наприклад,

1) T-Mobile в США стався великий витік даних (2012 рік);
2) атака на Microsoft Exchange Server (2021 рік);
3) крадіжка паролів користувачів eBay (2014 рік);
4) злам Dropbox у 2012 році, крадіжка паролів;
5) атака DAO та розкол мережі Ethereum;

6) кібератака на Yahoo! Наприкінці 2014 року;
7) кібератака на Sony Pictures у 2014 році;
8) злам LinkedIn у 2012 році;
9) вимагач WannaCry у 2017 році;
10) кібератака на Adobe у 2013 році;
11) Атака на мережу PlayStation у 2011 році;
12) кібератака на українську електромережу у 2025 році;
13) кібератака проти Естонії (2007 рік);
14) атака на сервери NASA (1999 рік);
15) вразливість у ПЗ MOVEit (2023 рік);
16) кібератака Petya.

2. Опишіть суть атакуючої кампанії, яку загрозу представляла атака, які методи використовувались для проникнення в систему (фішинг, зловмисне ПЗ, DDoS атаки, тощо). Розкрийте методи проникнення в систему:

1) фішинг: використання підроблених листів або вебсайтів для отримання особистих даних;
2) зловмисне ПЗ (malware): інфікування системи за допомогою вірусів, троянів, шпигунських програм;
3) атака DDoS: перевантаження системи великою кількістю запитів для відмови в обслуговуванні;
4) вразливості в програмному забезпеченні: використання невиправлених уразливостей для доступу до системи.

3. Опишіть цілі атаки, зазначте, що саме намагалися досягти зловмисники, наприклад, викрадення даних, фінансові втрати, саботаж, шантаж, порушення роботи системи.

4. Проаналізуйте наслідки атаки для постраждалої організації:

1) економічні втрати. Визначте, скільки коштувало організації відновлення після атаки (витрати на відновлення

даних, переривання бізнес-процесів, судові витрати, компенсації тощо);

2) вплив на репутацію. Оцініть, як атака вплинула на довіру клієнтів, партнерів та громадськості до організації. Це може включати втрату клієнтів, падіння акцій (якщо це публічна компанія), втрату довіри;

3) шкода для клієнтів і партнерів. Розкрийте, які конкретно збитки були нанесені клієнтам і партнерам (наприклад, витік особистих даних, скомпрометовані фінансові транзакції, порушення надання послуг).

5. Оцінити рівень захисту, який був реалізований в організації до атаки, та вказати, чому він не спрацював. Оцініть, які заходи безпеки були впроваджені в організації на момент атаки. Це можуть бути: антивірусне ПЗ, брандмауери, системи виявлення вторгнень (IDS/IPS), багатофакторна автентифікація, контроль доступу.

Які аспекти безпеки не спрацювали? Проаналізуйте, чому ці заходи не змогли запобігти атаці. Можливо, була виявлена вразливість у ПЗ, не було оновлень, недостатня обізнаність персоналу, чи просто невірно налаштовані системи безпеки.

6. Запропонуйте стратегії для покращення безпеки в аналогічних ситуаціях.

Визначте заходи, які можуть бути вжиті для запобігання подібним атакам у майбутньому. Це може бути використання передових систем захисту, оновлення ПЗ, регулярне сканування вразливостей.

6.1. Запропонуйте конкретні технічні стратегії для посилення безпеки, наприклад, налаштування брандмауерів, реалізація шифрування даних, запровадження принципу найменших привілеїв.

6.2. Порекомендуйте ввести програми регулярного навчання для співробітників з питань кібербезпеки, щоб знизити ризик фішингових атак та інших соціальних маніпуляцій.

6.3. Запропонуйте технології, які можуть бути корисними для організації в боротьбі з кіберзагрозами, такі як машинне навчання для виявлення аномальних дій або впровадження Zero Trust безпеки.

6. Підготуйте висновки щодо важливості аналізу реальних кейсів для підвищення кіберстійкості організацій.

Форма подачі завдання

Звіт (6-7 сторінок 14 шрифтом, 1.5 інтервал), презентація (5-6 слайдів). Звіт та презентація мають бути оформлені відповідно до академічних вимог, з посиланнями на джерела (якщо використовувались).

Індивідуальне завдання 2. Аналіз безпеки власних пристроїв

Мета завдання: провести аудит безпеки особистих пристроїв (смартфона та десктопного пристрою), оцінити стан їх захисту та розробити рекомендації для підвищення кібербезпеки.

Обґрунтування завдання: перевірка особистих пристроїв на наявність оновлень, антивірусного ПЗ та інших аспектів безпеки дозволяє ідентифікувати потенційні ризики, підвищити обізнаність користувача щодо налаштувань захисту та забезпечити захист персональних даних. У сучасних умовах кіберзагроз навіть мінімальні прогалини у безпеці можуть призвести до серйозних наслідків, включаючи витік даних, фінансові втрати та компрометацію облікових записів.

Завдання

1. Проведіть аудит безпеки власних пристроїв.

1.1. Виберіть два пристрої:

- смартфон (на базі Android або iOS);
- десктопний пристрій (Windows, macOS, або Linux).

1.2. Перевірте стан системи:

- наявність останніх оновлень операційної системи та встановлених програм;
- параметри автоматичного оновлення.

1.3. Оцініть встановлене ПЗ безпеки:

- наявність антивірусного програмного забезпечення;
- чи увімкнено брандмауер;
- використання VPN для захисту інтернет-з’єднань.

1.4. Перевірте конфігурацію доступу та автентифікації:

- наявність паролів або біометричної автентифікації;
- реалізацію багатофакторної автентифікації (MFA) для облікових записів;
- перевірку дозволів для встановлених додатків (доступ до камери, мікрофона, місцезнаходження).

2. Аналіз можливих ризиків.

2.1. Виявлення вразливостей:

- невстановлені оновлення або застаріле ПЗ;
- відсутність захисних механізмів (наприклад, антивірус або брандмауер);
- використання слабких паролів чи відсутність MFA;
- надмірні дозволи для додатків.

2.2. Оцінка ризиків для кожного пристрою:

- ймовірність витоку даних;
- можливість зараження зловмисним ПЗ;
- ризики компрометації облікових записів.

3. Запропонуйте рекомендації для підвищення безпеки:

3.1. Технічні заходи:

- регулярне оновлення ОС та програмного забезпечення;
- встановлення та регулярне оновлення антивірусного ПЗ;
- увімкнення брандмауера та VPN;
- перевірка дозволів для встановлених додатків.

3.2. Користувацькі заходи:

- вибір надійних паролів і налаштування MFA;
- обережність щодо відкриття посилань і завантаження файлів;
- уникнення використання незахищених Wi-Fi мереж.

4. Підготуйте висновки.

4.1. Оцініть, наскільки захищеними є ваші пристрої після проведення аналізу.

4.2. Визначте, які з виявлених проблем вимагають негайного втручання.

4.3. Сформулюйте загальні рекомендації щодо безпеки особистих пристроїв.

Форма подачі завдання

Звіт (10-15 сторінок 14 шрифтом, 1.5 інтервал), презентація (5-6 слайдів). У звіт включити скрини з двох пристроїв, які описуються (смартфон і настільний комп'ютер) і підтверджують кожен пункт завдання. Звіт та презентація мають бути оформлені відповідно до академічних вимог, з посиланнями на джерела (якщо використовувались).

Індивідуальне завдання 3. Аналіз безпеки браузера

Мета завдання: оцінити безпеку налаштувань веббраузера, виявити можливі ризики для конфіденційності та кіберзахисту, а також розробити рекомендації для покращення безпеки.

Обґрунтування: веббраузери є одним із найпоширеніших інструментів для доступу до інтернету, тому їхня безпека критично важлива. Налаштування браузера, такі як активність HTTPS, блокування трекерів і сторонніх файлів cookie, а також використання розширень, значно впливають на рівень захисту користувача. Аналіз налаштувань допоможе запобігти витоку даних, відстеженню та іншим ризикам.

Завдання

1. Проведіть аудит безпеки веббраузера.

1.1. Обрати веббраузер для аналізу, який ви використовуєте найчастіше. Наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, Opera або інший браузер.

1.2. Перевірити основні аспекти безпеки:

- HTTPS-з’єднання: впевнитися, що браузер автоматично перенаправляє користувача на захищені версії вебсайтів (HTTPS);
- блокування трекерів: перевірити, чи увімкнена функція захисту від стеження (наприклад, вбудовані функції браузера або розширення);
- контроль файлів cookie: оцінити політику браузера щодо зберігання та блокування сторонніх файлів cookie;

- налаштування конфіденційності: перевірити параметри збору даних браузером (телеметрія, автоматичне відправлення звітів).

1.3. Перевірте використання розширень:

- перегляньте встановлені розширення та оцініть їх безпеку (перевірте їхню репутацію, необхідність дозволів);
- виявіть, чи використовуються розширення для блокування реклами та захисту конфіденційності (наприклад, uBlock Origin, Privacy Badger).

1.4. Оцініть додаткові параметри:

- налаштування автоматичного оновлення браузера;
- використання менеджера паролів;
- наявність функцій ізоляції сайтів або інших засобів для запобігання атакам.

2. Аналіз можливих ризиків.

2.1. Виявлення вразливостей:

- використання незахищених HTTP-з'єднань;
- відсутність активного блокування трекерів;
- зберігання зайвих або небезпечних файлів cookie;
- використання ненадійних або надлишково привілейованих розширень.

2.2. Оцінка потенційних наслідків:

- можливість відстеження активності користувача трекерами;
- ризики витоку особистих даних через небезпечні розширення;
- загрози фішингових атак через недостатній захист від небезпечних сайтів.

3. Розробка рекомендацій.

3.1. Технічні заходи:

- увімкніть автоматичне перенаправлення на HTTPS (можна за допомогою розширень, таких як HTTPS Everywhere);
- налаштуйте блокування трекерів і сторонніх файлів cookie;
- перегляньте список встановлених розширень та видалити зайві чи ненадійні.

3.2. Користувацькі заходи:

- регулярно перевіряти налаштування конфіденційності браузера;
- використовувати менеджери паролів для уникнення збереження облікових даних у браузері;
- не завантажувати файли та розширення з неперевірених джерел.

3.3. Рекомендації щодо розширень:

- використовувати надійні розширення для блокування реклами (наприклад, AdBlock, uBlock Origin);
- встановлювати інструменти для покращення конфіденційності (Privacy Badger, Ghostery, VPN);
- обмежити дозволи для розширень (наприклад, доступ до історії браузера).

4. Підготуйте висновки:

4.1. Оцініть загальний рівень безпеки вашого браузера після аналізу.

4.2. Визначте, які зміни були внесені для покращення захисту.

4.3. Сформулюйте рекомендації для регулярного моніторингу безпеки браузера.

Форма подачі завдання

Звіт (10-15 сторінок 14 шрифтом, 1.5 інтервал), презентація (5-6 слайдів). У звіт включити скрини з браузера, які описуються і підтверджують кожен пункт завдання. Звіт та презентація мають бути оформлені відповідно до академічних вимог, з посиланнями на джерела (якщо використовувались).

Частина 4. Командні проекти

Командний проект 1. Безпека в цифровому світі: інструкція з кібергігієни для вразливих груп

Мета проекту: створити доступну та практичну інструкцію (пам’ятку) з кібергігієни для людей з вразливих соціальних груп (люди похилого віку, діти, люди з інвалідністю, люди з обмеженим доступом до цифрових технологій) з метою запобігання цифровим ризикам та покращити знання в галузі кібербезпеки.

Обґрунтування проекту: у сучасному цифровому світі вразливі групи населення піддаються високому ризику цифрових шахрайств, фішингу, персональних зловживань. Необхідно підвищити їхній рівень цифрової грамотності для забезпечення їхньої соціальної та фінансової безпеки .

Завдання

1. Формування команди (кількість учасників: 6-8 осіб). Розподіл на групи організовує викладач шляхом випадкового розподілу генератором випадкових чисел.

2. Визначіть, кого саме охоплює поняття "вразливі групи" у проєкті. Наприклад, це можуть бути діти, люди похилого віку або люди з обмеженим доступом до технологій. Відповідно для кожної групи свої рекомендації.

3. Розподіл ролей і обов'язків між учасниками.

Тут група обирає координатора, який розподіляє ролі (дослідники, контент-розробник, дизайнер) та забезпечує чіткість завдань, стежить за дотриманням етапів роботи (збір

даних, розробка тексту, дизайн, перевірка) презентує фінальну інструкцію та організовує відповідь на запитання від інших учасників чи викладача. Функції інших учасників:

а) дослідники: збирають інформацію про актуальні загрози кібербезпеки, які стосуються вразливих груп; аналізують заздалегідь підготовлені матеріали та обирають найбільш важливі теми для пам'ятки; пропонують рекомендації, які повинні увійти до інструкції; надають отримані дані контент-розробникам для створення тексту пам'ятки;

б) контент-розробник перетворює зібрану дослідниками інформацію на зрозумілі, чіткі та структуровані рекомендації; формує текст пам'ятки у простій і доступній формі, орієнтованій на вразливі групи (мінімум складної термінології); забезпечує логічну структуру інструкції (розділи, підзаголовки, марковані списки); оформляє текст для презентації і звіту; передає текст дизайнеру для подальшого оформлення та узгоджує правки;

в) дизайнер оформляє текст, додає відповідні ілюстрації на звіт і презентацію.

4. При підготовці рекомендацій із кібергігієни для вразливих груп важливо зосередитися на простоті та доступності інформації. Ось кілька основних аспектів, на які варто звернути увагу:

а) основи кібергігієни, зокрема основні принципи захисту персональних даних, таких як створення сильних паролів, уникання підозрілих лінків і файлів, а також важливість двофакторної автентифікації;

б) ризики та загрози, з якими можуть зіштовхуватися вразливі групи, наприклад, фішинг, соціальні маніпуляції, зараження шкідливим програмним забезпеченням або шахрайство;

в) прості та практичні поради такі як не використання однакових паролів на різних ресурсах, регулярне оновлення програмного забезпечення, виявлення та повідомлення про підозрілі активності;

г) контактна інформація або посилання на ресурс, де користувачі зможуть звернутися за допомогою у разі виникнення питань або потреби в додаткових інструкціях;

д) використання зрозумілої мови для людей з різним рівнем технічної підготовки, аби кожен зміг скористатися порадами незалежно від досвіду.

5. Обговоріть фінальні варіанти, внесіть корективи і презентуйте проект.

Форма подачі завдання

Інструкція з кібергігієни (10-15 сторінок 14 шрифтом, 1.5 інтервал), презентація (8-10 слайдів). Звіт та презентація мають бути оформлені відповідно до академічних вимог, з посиланнями на джерела (якщо використовувались). Презентацію і доповідь виголошує координатор групи.

Командний проект 2. Розробка плану реагування на кіберзагрозу

Мета проекту: розробка плану реагування на кіберзагрозу, створення чіткої та структурованої стратегії для оперативного реагування на потенційні кіберзагрози.

Обґрунтування проекту: розробка плану реагування на кіберзагрози є важливою для оперативного виявлення та нейтралізації загроз, що дозволяє знизити ризик серйозних фінансових, репутаційних та інших збитків. Чіткий план забезпечить швидке відновлення функціонування організації після кіберінциденту, мінімізуючи його вплив на бізнес-процеси.

Завдання

1. Викладач здійснює розподіл на групи генератором випадкових чисел.

2. Кожна самостійно організовує-розподіляє ролі кожного учасника.

3. Групам за жеребом пропонується один з вигаданих варіантів кіберінциденту, що відбувся:

3.1. Фірма "TechSecure" – займається розробкою охоронного обладнання. Співробітники компанії отримали фішингові листи, які виглядали як офіційні запити від постачальників, з метою отримання доступу до корпоративних облікових записів. Внаслідок цього деякі з облікових записів були скомпрометовані, що призвело до витоку конфіденційної інформації.

3.2. Фірма "AutoFix" – займається ремонтом автомобілів на всій країні. Один з серверів компанії зазнав атаки типу DDoS,

через що онлайн-послуги компанії були недоступні для клієнтів протягом кількох годин. Це викликало значні незручності для користувачів та збитки для бізнесу.

3.3. Фірма "BookStore Pro" – займається продажем книг по цілому світі. Під час оновлення програмного забезпечення виникла вразливість, яку зловмисники використали для проникнення в базу даних клієнтів. Внаслідок цього були викрадені персональні дані та платіжна інформація клієнтів, що спричинило втрату довіри до компанії.

3.4. Фірма "MediTech" – медична клініка. Один зі співробітників, який мав доступ до чутливих медичних даних, навмисно продав цю інформацію на чорному ринку. Це стало інсайдерською загрозою, яка серйозно піддала під сумнів безпеку персональних даних пацієнтів.

3.5. Фірма "SmartHome" – здійснює продаж техніки і програмного забезпечення для дому. Шкідливе програмне забезпечення проникло на систему управління розумними пристроями в будинках клієнтів через вразливість в одному з їхніх пристроїв. Зловмисники отримали контроль над усіма підключеними пристроями, що призвело до порушення приватності клієнтів.

3.6. Фірма "LogiWare" – займатися розробкою програмного забезпечення для управління логістичними процесами. Під час відновлення даних з резервної копії співробітник випадково завантажив застарілі копії, що призвело до втрати критичних даних про замовлення та клієнтів. В результаті постраждали як клієнти, так і репутація компанії.

4. Команди повинні описати покроковий план реагування, включаючи виявлення загрози, її ізоляцію та усунення, а також відновлення систем після інциденту. Команди повинні

враховувати моніторинг інциденту, повідомлення керівництва та клієнтів, а також проведення післяінцидентного аналізу для запобігання майбутнім атакам. У результаті проекту кожна команда повинна надати чіткий звіт з визначеними ролями та відповідальностями, а також конкретними діями для забезпечення безпеки.

Форма подачі завдання

План-реагування на інцидент (5-6 сторінок 14 шрифтом, 1.5 інтервал), інструкцію з запобігання подібних випадків (5-6 сторінок 14 шрифтом, 1.5 інтервал). Окремо зазначити внесок кожного учасника команди.

Командний проект 3. Проведення базового аудиту кібербезпеки організації

Мета проекту: провести базовий аудит кібербезпеки для вигаданої або реальної організації з метою виявлення вразливих місць, аналізу поточного стану безпеки та надання рекомендацій для покращення захисту від кіберзагроз.

Обґрунтування проекту: у сучасному світі кіберзагрози стають все більш складними та різноманітними, тому важливо регулярно проводити аудит кібербезпеки для виявлення слабких місць у системах організації. Це дозволить зменшити ризики втрати даних, фінансових збитків та репутаційних втрат.

Завдання

1. Формування команди (кількість учасників: 6-8 осіб). Розподіл на групи здійснюється за допомогою генератора випадкових чисел.
2. Команда обирає тип організації (реальну чи вигадану) для проведення аудиту кібербезпеки. Це може бути компанія, освітня установа, медична організація або інший тип бізнесу. Описати структуру організації та типи даних, які вона обробляє.
3. Учасники команди повинні обрати такі ролі: координатор, аналізатор загроз, оцінювач вразливостей, контент-розробник; дизайнер.
4. План аудиту:
    - провести огляд політик безпеки організації (паролі, доступ до даних, шифрування).

- перевірити безпеку мережі (файрволи, VPN, моніторинг трафіку).
- оцінити захист кінцевих пристроїв (комп'ютери, мобільні телефони, принтери).
- провести аналіз наявності шкідливого ПЗ та програм-вразливостей.
- оцінити рівень освіти співробітників щодо кібербезпеки (практики роботи з електронною поштою, паролями, інформаційними ресурсами).
- Перевірити наявність плану реагування на кіберзагрози.

Доповніть план аудиту ще двома пунктами і оцініть організацію за ними.

5. Розробіть рекомендації для поліпшення кібербезпеки.
6. Запропонуйте заходи щодо зміцнення фізичної безпеки.
7. Після виконання аудиту, команда повинна підготувати та представити фінальний звіт з рекомендаціями щодо покращення безпеки організації.

Форма подачі завдання

Звіт з аудиту кібербезпеки (10-15 сторінок, шрифт 14, 1.5 інтервал) з чітким викладом виявлених вразливостей та запропонованих заходів. Презентація (8-10 слайдів), яка підсумовує основні результати аудиту та рекомендації. Координатор презентує звіт та відповідає на запитання.

Командний проект 4. Розробка політики кібербезпеки з урахуванням юридичних вимог

Мета проекту: розробити комплексну політику кібербезпеки для організації, яка включає правила створення паролів і доступу до даних, порядок резервного копіювання, а також дії в разі кіберзагрози. Політика повинна відповідати чинним юридичним вимогам та регламентам, зокрема щодо захисту персональних даних.

Обґрунтування проекту: питання кібербезпеки важливим для захисту організаційних даних та забезпечення безпеки інформаційних систем. Водночас, організації повинні дотримуватись відповідних юридичних стандартів і законодавчих актів, що регулюють обробку персональних даних, захист інтелектуальної власності та боротьбу з кіберзлочинністю. Розробка політики кібербезпеки, яка враховує ці аспекти, допомагає мінімізувати правові ризики та покращити загальний рівень безпеки.

Завдання

1. Формування команди (кількість учасників: 6-8 осіб). Розподіл на групи здійснюється за допомогою генератора випадкових чисел.

2. Команда обирає тип організації (реальну чи вигадану) для розробки політики кібербезпеки. Описати структуру організації, типи оброблюваних даних, можливі кіберзагрози та основні юридичні вимоги, що можуть бути застосовані до неї

(наприклад, GDPR, Закон про захист персональних даних, стандарти ISO 27001).

3. Учасники повинні обрати такі ролі: координатор; юридичний експерт; аналізатор загроз; оцінювач вразливостей; контент-розробник; дизайнер.

4. Розробити політику кібербезпеки, враховуючи комунікація з правоохоронними органами та іншими зацікавленими сторонами з дотримання законів про захист персональних даних (наприклад, GDPR в ЄС, Закон України про захист персональних даних).

5. Перевірити відповідності політики безпеки законодавчим вимогам:

- аналіз юридичних аспектів політики кібербезпеки;
- визначення правил збору, зберігання та обробки персональних даних відповідно до місцевого та міжнародного законодавства;
- надання рекомендацій щодо дотримання вимог організаціям, які працюють в декількох юрисдикціях.

6. Після розробки документу, команда повинна презентувати політику кібербезпеки, підкреслюючи юридичні вимоги та запропоновані заходи.

Форма подачі завдання

Розроблена політика кібербезпеки (12-15 сторінок, шрифт 14, 1.5 інтервал) з чітким описом усіх аспектів, враховуючи юридичні вимоги та технічні заходи безпеки. Презентація політики кібербезпеки (8-10 слайдів), що підсумовує основні

моменти документу. Координатор презентує документ і відповідає на запитання.

Список використаних джерел

1. KeePassXC Password Manager [Електронний ресурс]. Режим доступу:https://keepassxc.org/.
2. Сервіс Pwned Passwords [Електронний ресурс]. Режим доступу: https://haveibeenpwned.com/Passwords.
3. Менеджер паролі Google [Електронний ресурс]. Режим доступу: https://passwords.google.com/.
4. Google Drive for Desktop Support [Електронний ресурс]. Режим доступу: https://support.google.com/drive#topic=14940.
5. Cryptomator [Електронний ресурс]. Режим доступу: https://cryptomator.org/.
6. Google Authenticator [Електронний ресурс]. Режим доступу: https://support.google.com/accounts/answer/1066447.
7. OpenSSl [Електронний ресурс]. Режим доступу: https://www.openssl.org/.
8. Email Header Analyzer [Електронний ресурс]. Режим доступу: https://mxtoolbox.com/EmailHeaders.aspx.
9. Online md5 generator [Електронний ресурс]. Режим доступу: https://www.md5.cz/ .
10. Hashcat [Електронний ресурс]. Режим доступу: https://hashcat.net/wiki/.
11. Mitmproxy [Електронний ресурс]. Режим доступу: https://mitmproxy.org/.

12. Alpine/bombardier [Електронний ресурс]. Режим доступу: https://github.com/alpine-docker/bombardier.
13. Nmap (Network Mapper) [Електронний ресурс]. Режим доступу:https://nmap.org/docs.html.
14. Ресурс habr.com [Електронний ресурс]. Режим доступу: https://habr.com/.