© Kireienko O. Attack scenarios on system of remote education // Ukrainian Scientific Journal of Information Security, 2024, vol. 30, issue 2, pp. 330-338. ©

СТЕГАНОГРАФІЯ ТА СТЕГОАНАЛІЗ / STEGANOGRAPHY & STEGANALYSIS

DOI: 10.18372/2225-5036.30.19246

СЦЕНАРІЇ АТАК НА СИСТЕМУ ДИСТАНЦІЙНОЇ ОСВІТИ

Олександр Кіреєнко

Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"

КІРЕЄНКО Олександр Володимирович, асистент
Рік та місце народження: 1993 рік, м. Київ, Україна.
Освіта: Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», 2016 рік.
Посада: асистент спеціальності «Кібербезпека» Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».
Наукові інтереси: інформаційна безпека, теорія ігор.
Публікації: більше 3 наукових публікацій, серед яких наукові статті.
E-mail: kirealex12@gmail.com.
Orcid ID: 0000-0001-9184-6738.

***Анотація**. Розробка моделей порушника та загроз необхідна для захисту інформаційної системи від потенційного шкідливого впливу. Шкідливий вплив на систему можуть чинити як випадково (її законні користувачі) так і з відповідним наміром (зловмисники). Кожна модель є абстракцією, а рівень цієї абстракції залежить від декількох критеріїв, одним із яких є об'єкт захисту. Системи дистанційної освіти застосовують при навчанні осіб в широкому віковому діапазоні – від школярів до повнолітніх студентів, а об'єктом атаки виступають навчальні матеріали та облікові записи користувачів, що містять їх персональні дані. Специфіка функціонування систем дистанційної освіти передбачає роботу з даними в різному форматі, а також інтерактивну, синхронну та асинхронну взаємодію користувачів, які при цьому можуть мати різні ролі в системі. При цьому варто очікувати і на зловживання основними функціями з боку законних користувачів, а також слідкувати за обмеженням доступу до системи із зовнішньої мережі.*

***Ключові слова:** система дистанційного навчання, водяний знак/watermark, стеганографія, формат даних, зворотній зв'язок. розділення секрету, пошук збігів, оптичне розпізнавання символів/OCR.*

Постановка проблеми

Система дистанційної освіти є інтернет-ресурсом, який надає користувачам можливість завантажувати, скачувати та переглядати файли із зазначеного заздалегідь діапазону форматів та чітко визначеного мінімального та максимального розміру для кожного із форматів, а також редагувати ці файли. Системи дистанційної освіти мають безліч додаткових функцій, кожна з яких створює додаткову поверхню атаки.

В першому наближенні системи дистанційної освіти подібні до соціальних мереж, але існують і відмінності, зумовлені обмеженням на кількість постів, які може зробити користувач, обов'язковим маркуванням часу завантаження та внесення змін, блокування можливості внесення змін після оцінювання і т.д. Оцінювання здійснюється викладачем чи групою викладачів, і допускається автоматичне оцінювання системою у випадках, коли надані відповіді добре формалізовані (наприклад представлені у вигляді тестів).

У випадках, коли студенту надається декілька спроб на виконання завдання, система може обрати найкращу із спроб, або останню із спроб. Використання середнього арифметичного суперечить педагогічним принципам, так як спонукатиме студентів одразу списувати правильні відповіді у своїх колег або попередників. Модуль сповіщення про надходження нових робіт від студентів може бути присутній, але не є обов'язковим, так як кількість сповіщень буде надмірною, а викладач і без нього перевіряє систему щодня.

Як і соціальні мережі, системи дистанційного навчання часто стають об'єктом атак, які можуть бути спрямовані як безпосередньо на систему, так і на її користувачів. Для захисту системи дистанційного навчання нам потрібно розуміти, які сценарії атак, крім загальних для веб-ресурсів, можуть бути в ній реалізовані.

Аналіз останніх досліджень і публікацій

На даний момент більше уваги приділяють захисту систем дистанційної освіти від зловживань із використанням засобів штучного інтелекту, таких як ChatGPT. Не можна забувати і про класичні загрози. Класичний підхід означає перевірку завантажуваних файлів на наявність шкідливого ПЗ, запобігання XSS атакам (в секції коментарів викладача під відповіддю студента), directory traversal для несанкціонованого

доступу до інших ресурсів сервера, на якому розміщено даний ресурс, створення резервних копій та балансування навантаження для запобігання атакам на відмову в обслуговуванні.

Специфіка функціонування системи дистанційної освіти полягає в її модульній структурі. Для коректного функціонування необхідно вирішити декілька непов'язаних одна з одною задач.

Технології водяного знаку необхідні для захисту навчальних матеріалів від поширення за межі системи. А для запобігання неправомірного використання матеріалів, витік яких все ж відбувся, потрібно впроваджувати механізми DRM.

Цифровий підпис може захистити відповіді студентів від модифікації, а для захисту навчальних матеріалів можна додатково застосувати стеганографію.

Для уникнення зловживань при перевірці відповідей потрібно застосовувати засоби розділення секретів. Механізм порівняння файлів потрібен для запобігання списуванню, але його бажано доповнити більш інтелектуальними засобами, такими як штучний інтелект. Для нетекстових даних також бажано мати модуль, що відповідає за оптичне розпізнавання символів.

Мета та постановка завдання

У цій роботі пропонується опис сценаріїв атак на систему дистанційної освіти, де крім класичних кібератак із використанням шкідливого ПЗ розглянуто широке коло порушень та зловживань пов'язаних із навчальними матеріалами та обліковими записами користувачів. Запобігання цим атакам дозволить знизити рівень матеріальних та репутаційних втрат та забезпечити високу якість навчання.

Виклад основного матеріалу дослідження

1. Потреба в спеціалізованій системі

Перш за все необхідно звернути увагу на те, що реалізувати дистанційне навчання звичайними, доступними кожному користувачу засобами не вийде. Викладач та студенти можуть використовувати для спілкування засоби електронної пошти або месенджер, файлові хостинги для зберігання навчальних матеріалів, а надіслані роботи зберігати локально. Але при такому підході виникає низка проблем. Жоден із цих ресурсів не розрахований на такий обсяг даних. Обсяг пам'яті, який виділено звичайному користувачу для збереження електронних листів буде дуже швидко вичерпано через сотні листів із вкладеними файлами. Так само складно зберігати відеофайли з записаними лекціями та допоміжні матеріали (такі як віртуальні машини з налаштованими операційними системами, середовища розробки під деякі мови програмування, СУБД та ін.)

Також великою проблемою буде прозорість навчального процесу в такому випадку. Студент повинен бути впевнений в тому, що отримані ним від викладача завдання є коректними, і такими ж за рівнем складності, як і завдання в інших студентів. Це не означає, що всі студенти повинні отримувати однакові завдання, але як мінімум вони повинні бачити список завдань розподілений по варіантам, а також відслідковувати стан перевірки своїх робіт, щоб не виникло ситуації, коли роботи одних студентів перевіряються одразу в день отримання, а інші чекають по 1-2 тижні.

Оцінювання теж повинно здійснюватися із підтримкою додаткових механізмів (наприклад перевірки на плагіат), та наданою викладачем аргументацією стосовно неправильних відповідей (щоб не було ситуації, коли студенту безпідставно занижують оцінки).

Окремо постає проблема захисту контенту. В країнах де навчання є платним, навчальні матеріали можуть бути скопійовані та викладені у відкритий доступ, де ними зможуть скористатися всі бажаючі, навіть якщо вони не поступили в даний університет по конкурсу або на контрактну форму навчання.

В даній статті описані загрози та атаки, пов'язані із ключовими елементами системи дистанційної освіти.

2. Користувачі системи

В загальному випадку можна виділити три категорії користувачів:

1) студенти;

2) викладачі;

3) адміністратори.

Ми не можемо використовувати мандатну модель розмежування доступу для нашої системи, тому що користувачі не знаходяться у відношення часткового порядку. Викладачі не мають права редагувати відповіді студентів, щоб не допомагати підставляти окремих осіб, а студенти не можуть редагувати оцінки викладачів. Тому обирати треба між дискреційною моделлю та рольовою. Основною перевагою рольової моделі є можливість використання групових ролей, що відповідає прикладній задачі, яку вирішує система.

Адміністратори відповідають за підтримку системи в робочому стані. Адміністраторам не обов'язково розбиратися в предметі, який будуть викладати. Саме адміністратори повинні реєструвати студентів та викладачів в системі. Викладач не повинен мати можливості створювати облікові записи для студентів, бо в цьому випадку викладач матиме можливість створити фіктивний обліковий запис для студента, який існує лише на папері. Ці "віртуальні" студенти будуть демонструвати надзвичайно високий рівень успішності, бо насправді від їх імені працює сам викладач. Це дозволить викладачу суттєво покращити статистику, збільшивши кількість відмінників, що може впливати на його особисті показники. Або навпаки - створити багато студентів із незадовільними результатами, щоб підставити свого колегу-лектора, який неналежним чином викладає матеріал(в цьому випадку не доброчесним є викладач, що відповідає за практичні/лабораторні роботи).

Так само не повинно бути можливості реєструватися в системі самостійно у самих студентів. До курсу дистанційного навчання повинні допускати лише тих осіб, які офіційно навчаються в нашому навчальному закладі. Але проблема не лише в тому, що з курсом може ознайомитися стороння особа. Студент може створити фіктивний обліковий запис для пробного проходження практичних/лабораторних робіт, для виявлення помилок, щоб пізніше переробити

дані завдання вже без помилок з основного свого акаунта. Також зайві облікові записи можуть бути використані для створення додаткового навантаження на викладача, особливо перед сесією, щоб спонукати викладача перевіряти надіслані роботи швидше, а значить і менш ретельно, збільшуючи імовірність того, що викладач не помітить якусь помилку.

Крім реєстрації користувачів в системі, потрібно передбачити і протилежну дію - видалення облікового запису. Видаленням облікового запису теж можна зловживати. Видаливши обліковий запис студента можна створити йому труднощі, примусивши переробляти практичні/лабораторні роботи, щоб відновити втрачені дані. Видалення облікового запису викладача може створити ще більше проблем, так як без цього користувача в системі не буде можливості читати лекції, перевіряти надіслані завдання. Обліковий запис викладача може стати об'єктом атаки зі сторони студента із низькими оцінками для того щоб відтермінувати здачу робіт, додавши собі декілька годин або днів на виконання завдань, коли без подібної атаки він не встигає до дедлайну. Навіть якщо студент зберігає всі виконані завдання локально, йому все ще може знадобитися час для перенесення відповідей з персонального комп'ютера в систему дистанційного навчання. Навіть якщо всі відповіді будуть в текстовому форматі, вони все ще можуть відображатися в системі із значною затримкою, якщо в системі реалізована попередня перевірка на плагіат/списування, а також інші порушення. Якщо для економії часу викладача система не відображає надіслані студентами завдання до моменту, коли для цих завдань буде підтвердження у відсутності плагіату, то видалення облікового запису студента із подальшим його відновленням і перепроходженням всіх завдань може зайняти більше часу, ніж просте "механічне" перенесення відповідей із текстового файлу на комп'ютері студента в форми для відповідей в додатку системи дистанційного навчання. (аналогічно для випадку із завантаженням файлів на сервер такої системи).

Облікові записи студентів повинні бути захищені від інших студентів. Якщо у студента виникне конфлікт з іншим студентом, можливою є ситуація, коли цей другий студент спробує отримати доступ до облікового запису жертви для руйнування її рейтингу. Це може бути цілеспрямоване проходження тесту, або виконання лабораторних/практичних робіт з дуже низькою якістю, або внесення відповідей, що будуть розпізнані системою як плагіат. Тому розподілом паролів до облікових записів студентів повинен займатися спеціально призначений працівник. Викладач цього робити не може, бо тоді викладач знатиме паролі студентів, що дозволяє від їх імені виконувати завдання на занадто високий/низький бал. Для уникнення цієї проблеми можна передбачити функцію зміни паролю користувачем після першого входу в обліковий запис.

Також необхідно передбачити захист від випадкового видалення облікового запису. При видаленні запису він спочатку блокується, а саме видалення відбувається лише через деякий фіксований час. Це дозволить відновити обліковий запис у випадку, коли виникне така потреба.

Так як в нашій системі користувачі мають чітко визначені ролі і переключатися між ролями їм не треба, ми можемо замість однієї загальної множини користувачів $U_$заг, використати підмножини користувачів студентів U_s, викладачів U_t та адміністраторів U_a. Навіть якщо хтось із студентів закінчить університет і стане викладачем, перенесення його із множини студентів до множини викладачів відбуватиметься в літній період, коли система не використовується, тому це не порушить роботу системи. А от перехід користувача із однієї множини до іншої протягом навчального року буде одразу свідчити про атаку, так як хтось із студентів намагається стати викладачем, щоб перевірити свої ж роботи, або роботи друзів на дуже високий бал, або у випадку із конфліктом з іншим студентом – на дуже низький бал:

$$U\_s' = U_s \backslash u_j,$$
$$U\_t' = U_t \cup u_j.$$

Всіх студентів із множини U_s ми переносимо у відповідні навчальні групи. Можливою є ситуація, коли один студент одночасно знаходиться в декількох групах. Це можна спостерігати тоді, коли студент має факультативні дисципліни, що не є обов'язковими, або отримує дуальну освіту. Частка таких студентів дуже мала, а тому подібні збіги можна перевіряти вручну, але перевіряти їх все ж потрібно, тому що включення студента до групи надає йому доступ до навчальних матеріалів по тим предметам, з якими працює група. Поширення навчальних матеріалів за межі системи є атакою і нам потрібно точно знати, хто мав до них доступ. Ще однією проблемою є ідентифікація студентів у яких однакове прізвище та ініціали. Для уникнення проблем нам недостатньо використовувати прізвище студента з його повним ім'ям та побатькові, так як це не вирішує проблему із повними тезками. А тому всі студенти повинні бути однозначно ідентифіковані за допомогою додаткової інформації. У випадку із студентами вищих навчальних закладів це може бути номер залікової книжки. Для школярів теж можна запровадити подібний ідентифікатор.

Таким чином об'єкт, що представляє студента буде представлено кортежем вигляду

$u_i =< FN, LN, PN, \boldsymbol{ID}, \dots >$, *де*

FN – first name, LN – last name, PN – parent name,

ID – згаданий вище ідентифікатор.

Кожна група студентів має доступ до деякої підмножини дисциплін (в системах дистанційної освіти може використовуватися термін "курс")

$D_{G\_i} = \{D_{i1}, D_{i2}, \dots, D_{ik}\}$, де

D_{G_i} – дисципліни/курси, до яких додали і-ту групу

Так само кожна дисципліна/курс відкрита для деякої підмножини груп

$G_{D\_i} = \{G_{i1}, G_{i2}, \dots, G_{im}\}$, де

G_{D_i} - групи, що мають доступ до і-тої дисципліни

Для економії ресурсів можна зберігати лише один із зазначених наборів, але це суттєво знижує спостережність системи.

Для спостережності і простоти адміністрування людиною краще зберігати ці дані у вигляді матриці,

де по горизонталі відмічені дисципліни, а по вертикалі групи:

$$GD = \begin{pmatrix} gd11 & \cdots & gd1n \\ \vdots & \ddots & \vdots \\ gdm1 & \cdots & gdmn \end{pmatrix}$$

де $gd_{ij} = 1$, якщо і-та група вивчає j-ту дисципліну, та 0 у протилежному випадку.

Якщо систему дистанційного навчання розробляють для школи, матриця GD матиме особливий вигляд.

За умови відсутності додаткових факультативних предметів, кожен рядок буде представлений у вигляді послідовності нулів, після якої слідує послідовність одиниць, після якої знову слідує послідовність нулів. Для перших та випускних класів будуть відсутні перша та остання послідовність нулів відповідно:

$$gd_i = [gdi1, \dots, gdi\ t-1, gdit, \dots, gdi\ t+k, gdi\ t+k+1, \dots gdi\ n],$$

$$gdij = \begin{cases} 0, j < t \\ 1, t \le j \le t+k \\ 0, t+k < j, \end{cases}$$

це пояснюється тим, що в школах учні вивчають всі предмети, які відповідають деякому року навчання, тому для 6 класу в таблиці будуть стояти 0 для предметів 1-5 класів, одиниці для 6 класу і нулі для 7-12 класів. В самому кінці таблиці можуть зберігатися факультативні предмети. Якщо при цьому ми ще і відсортуємо таблицю по стовпцям, то подібна умова буде виконуватися і для стовпців. Наприклад до предмету математика 1 класу матимуть доступ класи 1А, 1Б, 1В, в той час як всі старші класи (2А, 2Б....12Б, 12В) не будуть мати жодного відношення до даного предмету. Тобто подібна матриця складатиметься із прямокутних блоків із одиниць на діагоналі, а решта буде заповнена нулями. При такому представленні дуже легко помітити групи,яким надали неправомірний доступ до навчальних дисциплін, та групи, яким забули надати доступ до обов'язкових дисциплін.

Аналогічно можна задати і співвідношення для дисциплін та викладачів, які за них відповідають.

Однією із найбільш поширених причин затримки при здачі робіт є хвороба і відповідний візит до лікаря. Студентів не можна покарати за форс мажор, яким є хвороба, але посувати термін здачі робіт у відповідь на будь-яку заяву про погане самопочуття ми не можемо. Як при очному, так і при дистанційному навчанні для підтвердження хвороби використовується довідка від лікаря. Ця довідка містить інформацію про стан здоров'я людини і вона однозначно пов'язана із людиною за ПІБ. Тому дана довідка відноситься до категорії персональних даних, які необхідно захищати. Найбільш критичним звісно буде захист персональних даних неповнолітніх студентів, так як для цього нам знадобиться додатково взаємодіяти із їх батьками чи опікунами. Найкращим способом захисту таких даних є їх видалення із системи, як тільки рішення про перенесення дедлайну здачі деякої роботи для студента, який хворів, прийнято. Оптимальним варіантом є їх завантаження в чітко визначений часовий інтервал, коли перевіряюча сторона (викладач) очікує їх побачити, прийняття рішення про перенесення дедлайну і миттєве видалення. Тобто викладача завчасно повідомляють, коли буде завантажено довідку, щоб він в цей час був в системі і зміг видалити її якнайшвидше. Якщо потім знадобиться перевірити дані, довідку можна буде завантажити повторно. Довідка не повинна зберігатися в системі з вечора п'ятниці до ранку понеділка. Тривале знаходження цих даних в системі збільшує вікно можливостей для порушника. Хоча і при короткотривалому зберіганні в системі може відбутися витік даних.

3. Навчальні матеріали

Система дистанційного навчання передбачає як інтерактивну, так і асинхронну взаємодію між викладачами та студентами. Всі лекції повинні записуватися і зберігатися для подальшого включення їх (в повному вигляді або фрагментами) до навчальних матеріалів асинхронної взаємодії. Це дозволить в перспективі знизити навантаження на лектора, так як замість проведення он-лайн заняття можна буде ввімкнути запис лекції. Також це дозволяє вирішувати спірні питання, коли лектор допустив помилку під час лекції, а студент у зв'язку із цією помилкою неправильно виконав лабораторну/практичну роботу. Запис лекцій також дозволяє відслідковувати порушення під час лекцій як з боку лектора, так і з боку студентів.

Усі навчальні матеріали можна розділити на 3 категорії:

1) недоступні для скачування;

2) доступні для скачування лише в деякий час;

3) доступні для скачування без обмежень.

Якщо матеріали недоступні для скачування, то студенти можуть взаємодіяти з ними лише засобами системи дистанційного навчання. Система може відслідковувати програмне забезпечення для скріншотів та запису відео з екрану, але не може відслідковувати апаратне забезпечення за межами клієнта. Студент все ще може сфотографувати екран за допомогою смартфону. Аналогічно можливим є запис відео з екрану. Це не означає, що засобів, які б запобігали скачуванню, або хоча б відслідковували подібні дії, бути не повинно. Якщо ми не можемо заборонити студентам знімати відео з екрану на стороні клієнта, ми все ще можемо генерувати watermark, що динамічно змінюється, і є унікальним для кожного студента в системі. Таким чином, якщо матеріали, що не підлягали завантаженню, будуть викладені у відкритий доступ, то за наявним на відео watermark можна буде встановити, хто саме із студентів виконав запис цих матеріалів.

Недоліком подібного підходу є те, що студент може під час запису закрити рукою/папірцем/ще чимось частину екрану, на якій розміщено watermark. Це дозволить йому викласти подібні матеріали в мережу анонімно. Захист матеріалів від перегляду їх людьми, в яких не повинно бути доступу до курсу, відбувається в режимі "гонки озброєнь ", коли студенти придумують нові методи запису таких матеріалів. Нехай замість динамічного watermark ми зробили watermark, який постійно пересувається по екрану. Тобто студент вже не зможе просто закрити його долонею чи папірцем. Але студент все ще може використати програмне забезпечення для роботи з відео і вирізати watermark з кожного кадру. Для цього знадобиться більше зусиль, але це зробити можливо. Нехай

тоді весь екран буде захищений watermark. Тепер навіть невеликий фрагмент екрану з деякої послідовності кадрів дозволить виявити автора запису. В такому випадку студент може просто переписати матеріали із відео в конспект або презентацію powerpoint і викласти у відкритий доступ вже її. Використовуючи самого себе в якості проміжної ланки при копіюванні матеріалів, студент може видалити будь-який watermark. Але є і більш простий підхід. Якщо бажання злити матеріали у відкритий доступ виявило декілька студентів, вони можуть записати відео з екрану, а потім використати простий скрипт, який скомпонує нове відео, використовуючи кадри із декількох однакових відео, що відрізняються лише watermark. В результуючому відео динамічний watermark буде відрізнятися від watermark кожного студента, тому що кадри відео беруться із різних версій. Таким чином, якби в одному відео було закодовано послідовність <1,2,3,4,5> а в іншому <6,7,8,9,10>, то в результуючому відео, де кадри беруться по черзі з обох, послідовність буде мати вигляд <1,7,3,9,5> і цю послідовність не можна буде прив'язати вже до жодного із студентів. При цьому студенти можуть використовувати і більш складний варіант компонування відео, а не просте чередування кадрів. Кадри можуть обиратися випадковим чином. І чим більше відео використовується при компонуванні, тим більше інформації повинно міститися в кожному кадрі, щоб визначити, якому саме студенту належав кадр із відзнятого відео. Для того, щоб встановити студентів, відповідальних за викладення такого скомпонованого відео, нам потрібно, щоб кожен студент отримував унікальну послідовність закодованих у watermark символів. В цьому випадку будь-який символ буде однозначно вказувати на одного конкретного студента. Але це означає, що Watermark буде дуже перевантаженим і суттєво впливатиме на якість зображення. Також пам'ятаємо, що студенти все ще залишають за собою можливість просто перенести матеріали із відео в нове відео або презентацію в форматі powerpoint і викласти у відкритий доступ уже цей новий файл.

Для захисту навчальних матеріалів від поширення в мережі можна використати інший підхід - *захист формату даних*. Для цього нам потрібно зберігати всі матеріали в форматі, який може розпізнавати лише наша система дистанційного навчання. Скачування подібного файлу саме по собі не є загрозою, якщо у користувачів нема програми, яка б могла прочитати такий файл. Ми можемо використовувати нестандартні формати для збереження тексту, аудіо, відео та інших файлів. Сама ж система для дистанційного навчання може просто не мати функції копіювання тексту. Тобто при спробі виділити текст на екрані і скопіювати його в блокнот, у користувача або взагалі нічого не скопіюється, або скопіюється текст із кодуванням і в блокнот (або інший текстовий редактор) скопіюються "випадкові" символи із ASCII таблиці. Студент все ще може переписати текст вручну, тому цей підхід не є надійним на всі 100%. Але він принаймні захищає від копіювання великі обсяги даних, які вручну переписувати довго. Але і тут захист відбувається в форматі "гонки озброєнь ". Студенти можуть записати відео, де вони "пролистують" підручник, використовуючи засоби системи дистанційного навчання для коректного відображення даних, і викласти у відкритий доступ вже це відео. Також вони можуть нарізати дане відео на кадри і використати засоби розпізнавання тексту, щоб з кожного кадру витягнути текст. Таким чином навіть великі обсяги даних можуть бути скопійовані без нашого дозволу. Для захисту від розпізнавання символів, ми можемо використовувати якийсь нестандартний шрифт або додати "шум" на кожну сторінку, щоб система розпізнавання не могла виділити на сторінці текст. Але це все ще можуть зробити студенти, і чим більше їх, тим більший обсяг інформації вони можуть перенести із системи дистанційного навчання у відкритий доступ, просто розподіливши обов'язки порівну між собою.

Якщо навчальні матеріали написані мовою, що не належить до мов міжнародного спілкування, витік інформації відбуватиметься повільніше, так як спочатку матеріал потрібно буде перекласти. Як і в попередніх випадках, студенти-порушники будуть намагатися автоматизувати процес перекладу, щоб перенести матеріали із системи дистанційного навчання у відкритий доступ швидше.

Матеріали, які повинні бути доступні студентам лише протягом деякого часу (наприклад протягом проходження курсу) потрібно захищати так само, як і матеріали попередньої категорії (бо тимчасово доступними вони повинні бути лише для студентів, а не для сторонніх осіб, які взагалі в нашому закладі не навчаються), і додатково передбачити їх блокування чи видалення після деякого дедлайну. Налаштувати автоматичне видалення файлів або блокування доступу до них в певний момент часу не складно. Це можна зробити і за допомогою утиліти Crontab в лінуксі, але всі ці механізми захисту не будуть поширюватися на створені студентами копії. Будь-які файли повинні відображатися лише через інтерфейс додатку, який при цьому перевіряє поточну дату та інші умови. Поточну дату потрібно перевіряти не за локальним часом, бо студенти можуть виставити довільне значення на своєму персональному комп'ютері, а через Інтернет. При цьому студенти можуть використати wireshark для перехоплення пакетів і модифікації відповіді від сервера, щоб переконати нашу систему в тому, що файли повинні все ще бути доступними. Примусити студента видалити файл, який було скачано, практично неможливо. Тому файли не повинні скачуватися окремо. Студент повинен для взаємодії із матеріалами курсу скачати весь додаток, в якому вже відображаються навчальні матеріали. І додаток буде відповідальним за видалення файлів при настанні деякої умови.

Уповільнити поширення матеріалів першої та другої категорії можна, зробивши їх більшими. Штучно нарощуючи розмір файлу ми збільшуємо час необхідний на його завантаження/скачування, а також збільшуємо витрати на його зберігання.

Для швидкої передачі інформації студенту потрібно буде самостійно конвертувати інформацію навчального курсу у файл стандартного розширення і меншого розміру. Даний підхід є незручним для студентів, які сплачують за послуги Інтернет провайдера

пропорційно до трафіку, а не по безлімітному тарифу на місяць.

4. Надсилання робіт

В системі дистанційного навчання мають бути передбачені механізми прийому надісланих студентами робіт. Це можуть бути лабораторні, практичні роботи, есе, тести, курсові роботи, відповіді на екзаменаційні білети та ін.

Завантаження робіт на сервер системи дистанційного навчання також пов'язане з суттєвими ризиками. Студенти ПОВИННІ мати можливість завантажувати інформацію на сервер в деякому форматі, бо інакше вони не зможуть здавати виконані під час навчання роботи. Але ми не можемо допускати завантаження довільних файлів в довільні моменти часу та довільну кількість спроб. Деякі завдання передбачають дедлайни, і надсилання роботи після дедлайну повинно або блокуватися, або оцінка за таку роботу повинна автоматично знижуватися. Очевидно, що студенти, що надсилають роботи пізніше, матимуть бажання обійти обмеження по дедлайнам для здачі робіт. Вони можуть для цього маніпулювати метаданими файлу, змінюючи його дату створення, щоб робота виглядала так, ніби вона була виконана раніше. Сервер може додавати власні метадані до всіх завантажених файлів для уникнення такої проблеми. Студенти також можуть надіслати пошкоджений файл, щоб "виграти час ", бо у випадку із розміщенням пошкодженого файлу в системі може існувати "презумпція невинуватості " студента, відповідно до якої, файл пошкоджений не з вини студента і не є спробою виграти час.

При завантаженні великих файлів потрібно враховувати не момент створення копії файлу на сервері, а момент початку завантаження. Це необхідно для того, шоб недоброчесний студент, який списав роботу в іншого студента не міг завантажити її швидше, ніж оригінал. Потрібно пам'ятати, що в нашій системі може існувати перевірка на дублікати і тоді найбільш ранній файл признається оригіналом, а решта будуть вважатися списаними.

Ще одним поширеним порушенням при здачі робіт є підміна файлу з відповіддю, що вже була оцінена (і скоріше за все як неправильна). Якщо студент надсилає не сам файл, а посилання на гугл-диск, або якийсь інший хостинг, то в нього залишається можливість редагувати файл, на який вказує відповідне посилання. Це може призвести до ситуації, коли викладач чесно оцінив надіслану роботу як неправильну, після чого студент в односторонньому порядку поміняв текст відповіді і тепер звинувачує викладача в недоброчесності. Ця ситуація є критичною для всієї системи дистанційного навчання, бо вона дозволяє обходити механізм оцінювання. Очевидним рішенням проблеми було б завантаження самих файлів, а не посилань на них, але подібне рішення не працює для файлів великих розмірів. Крім виправлення неправильної відповіді на правильну, можлива і протилежна ситуація, коли вже оцінену на високий бал роботу змінюють, щоб вона виглядала погано. Це також може бути використано для зниження репутації викладача, бо все виглядатиме так, ніби викладач поставив оцінку незаслужено.

Для деяких робіт може бути передбачена деяка кількість спроб. Одразу ж потрібно вирішити, як обирається оцінка – за останню спробу, чи за кращу спробу. Яким би підходом ми не скористалися, проблема з таким типом взаємодії із системою дистанційного навчання в тому, що студент може знайти правильну відповідь методом виключення, або просто вгадати її з декількох спроб. В сфері навчання значення має не оцінка, а знання, яким вона відповідає. Якщо студент не може аргументувати вибір відповіді, вона не повинна зараховуватися навіть якщо відповідь правильна. Тому для більшості подібних робіт повинно бути передбачено поле для коментарів зі сторони студентів, де вони можуть аргументовано захистити свою відповідь.

Виконання лабораторної роботи в межах однієї спроби може бути перервано з технічних причин (наприклад зникло інтернет-з'єднання або взагалі світло). В цьому випадку потрібно переконатися, що спроба не "згорить". З іншого боку, надання студенту ще однієї спроби у відповідь на технічні несправності в поточній спробі стимулюватиме недоброчесних студентів створювати нештатні ситуації, щоб їм надали ще одну спробу.

5. Оскарження оцінки

Коли студент надсилає лабораторну чи практичну роботу, що була виконана не правильно (зараз розглядаємо саме варіант помилки, а не плагіат/списування), він повинен бути повідомлений про наявність помилки. Проблема в тому, що студенту не завжди можна повідомити, *в чому саме полягає помилка.* Бо в деяких випадках це означає відкрито оголосити йому правильну відповідь. Якщо завдання, в якому було допущено помилку, не є індивідуальним, то розголосивши правильну відповідь ми розголошуємо її всім студентам, які зараз навчаються на цьому курсі, а також всім студентам, які будуть навчатися після них і так до тих пір, поки дане питання взагалі не замінять. Навіть якщо студенти отримують завдання по варіантам, це все одно означає, що студенти, які проходитимуть даний курс в наступному році, через два роки і т.д. матимуть правильну відповідь на питання в лабораторній/практичній роботі для деякого варіанту. А так як різні студенти мають різні варіанти і допускають в них помилки, то досить швидко вони можуть зібрати повний набір правильних відповідей. Для цього достатньо, щоб в кожному варіанті завдань помилку допустив хоча б один студент. Після цього він отримує правильну відповідь від викладача і далі цю відповідь студенти наступних років навчання будуть просто копіювати. Це суттєво впливає на якість навчання, і допускати подібну ситуацію не можна.

Не повідомляти студенту, в чому полягає помилка, теж проблематично, бо в цьому випадку студент починає сумніватися в доброчесності викладача. Цілком закономірною є ситуація, коли студент почне думати, що йому спеціально занижують оцінку, і що відповідь була насправді правильною. На жаль у даної проблеми немає простого рішення. Ми не можемо змінювати набір питань для студентів щороку просто тому, що деякий матеріал їм потрібно вивчити в будь-якому разі і цей матеріал буде однаковим кожного року. Але нам потрібно гарантувати, що всі претензії

студентів будуть відкинуті по завершенню курсу. В нашій системі дистанційного навчання повинен бути передбачений механізм, за яким ми все ж повідомляємо студенту правильну відповідь в кінці семестру, коли предмет вже здано, якщо до цього моменту він так і не знайшов правильну відповідь. Але просто повідомити правильну відповідь не достатньо. Студент, отримавши відповідь від викладача, все ще може вважати, що причина, з якої завдання було не зараховане, була вигадана викладачем в останній момент, а протягом семестру йому просто створювали перешкоди під час захисту лабораторних/практичних робіт. На відміну від попередньої проблеми, цю ми можемо вирішити. Коли студент отримує незадовільну оцінку, ми можемо відправити йому список його помилок, навіть якщо такий список розголошує правильну відповідь, але даний список повинен стати доступний для читання лише після завершення семестру. Тобто в кінці семестру студент повинен отримати доступ до зауважень, які виникли до його робіт, і переконатися, що ці зауваження справді були коректними і актуальними протягом всього семестру, а не були вигадані викладачем в останній момент. Застосувати підхід розділення секретів за допомогою операції XOR ми не можемо, тому що будь-яка надана студенту послідовність символів може бути перетворена на будь-яку іншу, якщо її скласти із відповідною послідовністю символів за допомогою операції XOR. Тобто при розділенні секретів цим способом, недоброчесний викладач все ще може вигадати причину в останній момент, що є неприйнятним для студента. Набагато кращим є варіант, при якому студент отримує список із зауваженнями до своїх робіт у вигляді архіву із встановленим паролем. Так як архів можна відкрити лише при наявному паролі, розшифрувати відповідь в довільний текст, придуманий в останній момент, викладач вже не зможе. Недоліком такого підходу є те, що студент може спробувати відкрити архів до завершення семестру, скориставшись засобами перебору паролів, наприклад утилітою *john the ripper*. Пароль на архіві повинен бути досить складним, щоб протягом семестру його не вдалося знайти методом перебору. Ще одним варіантом вирішення даної проблеми є використання засобів асиметричної криптографії. Студент отримує список зауважень у вигляді зашифрованого повідомлення, до якого додано цифровий підпис. Студент може перевірити, що цифровий підпис є дійсним, навіть якщо підписаний файл зашифрований. В кінці семестру студент отримує ключ для розшифрування файлу. Підмінити відповідь на "вигадану в останній момент " практично неможливо, так як для цього викладачу потрібно знайти такий ключ K при якому шифроване повідомлення C1 розшифровується не в оригінальний (і осмислений) відкритий текст P1, а в інший відкритий і осмислений відкритий текст P2, який при цьому виглядає як список зауважень. При достатній довжині ключа підбір паролю буде практично неможливим (необхідний час на декілька порядків більший від тривалості семестру). Недоліком звісно буде необхідність розголошення секретного ключа, який використовувався для шифрування. Навіть якщо використовувати один і той самий ключ для всіх студентів (їх спільних зусиль все ще недостатньо для підбору паролю за прийнятний час), нам все одно потрібно буде змінювати пароль 1 або 2 рази на рік (залежить від того, чи викладають дану дисципліну в кожному семестрі). І, як уже зазначалося вище, це ніяк не захищає нас від розголошення цих відповідей студентам, які навчатимуться в наступному році.

6. Виявлення списування

Проблема, що була описана в попередньому розділі не має загального розв'язку, але один механізм захисту ми все ж таки можемо впровадити. Для того, щоб усі студенти, які навчатимуться в наступному році, не скористалися відкритими списками зауважень, що були отримані студентами поточного року, нам потрібно поширити серед студенів декілька таких списків, серед яких лише один є коректним, а в решті цілеспрямовано допущені помилки. Списування відповідей із одного із таких списків може бути легко виявлено, а до студента застосоване відповідне покарання. Помилки можуть бути як смислові (відповідь, що є просто неправильною) так і граматичні/орфографічні (зайва кома чи літера). Особливо зручними для викладача будуть помилки, які з'являються при застосування google translate. Якщо студенту необхідно в контексті однієї лабораторної/практичної роботи написати декілька відповідей, то в некоректному списку можна пропустити одну відповідь, або дописати одну зайву. В такому випадку при копіюванні відповідей в сліпу, у студента всі відповіді міститься на одну і буде видно невідповідність питання та відповіді до нього. Дані підходи не дають 100% гарантії виявлення списування, так як уважний студент може помітити всі ці недоліки, але для цього йому краще розбиратися в матеріалі курсу, що є кінцевою метою навчання, як дистанційного, так і очного.

Наступним важливим механізмом є підсистема, що аналізуватиме відповіді студентів на збіги. Деякі питання можуть передбачати "однаковість" правильної відповіді, при якій ми не будемо підозрювати студента в списуванні. Наприклад це стосується питань, що містять математичні приклади та питань, що потребують дати чітке означення деякому терміну. В цих випадках ми здійснюємо перевірку не для того, щоб виявити списування, бо скоріше за все студенти просто переписують текст із конспекту, який був попередньо продиктований лектором, а для того, щоб оцінка однакових відповідей була однаковою.

Якщо система здатна виявити однакові відповіді, то вона ж може без участі викладача проставити автоматично однакові оцінки за такі відповіді. Це суттєво прискорює процес перевірки деяких лабораторних/практичних робіт. При цьому система повинна порівнювати інформацію, що міститься в відповіді, а не сам файл із відповіддю. Наприклад для лабораторної роботи з предмету "Операційні системи ", наша система дистанційної освіти повинна перевірити, щоб введені користувачем в термінал команди і вивід терміналу у відповідь на ці команди співпадав. Ім'я користувача та назва дистрибутиву, що відображаються в терміналі, можуть при цьому відрізнятися, але відповідь все одно вважається однаковою. Представлення відповідей студентів у вигляді текстового файлу в цьому випадку робить таку перевірку досить

простою, але створює вразливість, при якій студент може просто відредагувати текстовий файл, змінивши відповіді ОС, які виводилися в термінал на "правильні". Якщо замість текстового файлу студенти завантажують в системи скріншоти з терміналом, то для досягнення цього ж результату знадобляться засоби OCR.

7. Робота в групі

Як очне так і дистанційне навчання вже давно перестали бути виключно індивідуальним. Студенти протягом семестру будуть отримувати завдання, які необхідно виконати в складі групи. Під групою тут потрібно розуміти не всю навчальну групу із 30+ осіб, а команду із 2-5 осіб, що працюватимуть над спільним проектом. Розділити множину студентів на декілька підмножин приблизно рівного розміру не складно. Розділити множину студентів на декілька підмножин з урахуванням їх побажань – задача, яка може взагалі не мати розв'язку. Навіть якщо ми будемо ігнорувати позитивні побажання (бути розподіленим у групу із відмінником), і враховуватимемо лише чорні списки (небажання окремих студентів працювати одне з одним), ця задача все ще може викликати суттєві затримки в роботі системи. В найпростішому випадку нам потрібно розбити студентів на пари, тобто команди із 2 осіб. В цьому випадку ми можемо представити множину студентів у вигляді графу G, вершинами якого є студенти, а ребрами – зв'язки, що вказують на бажання студентів працювати одне з одним. Тобто в такому графі між парою вершин u_i, u_k існує ребро тоді і тільки тоді, коли і-тий та j-тий студенти не знаходяться в чорних списках одне одного. Задача формування команд в цьому випадку зводиться до перетворення графу $F(G) = G'$, яке застосовується ітеративно так, що $G' = G \backslash \{u_i, u_j\}$, доки не залишиться порожня множина. Якщо матриця суміжності графу G має розмірність nXn то матриця суміжності графу G' матиме розмірність $(n-2)X(n-2)$. Існує C_n^2 способи обрати 2 елементи із n, після чого потрібно вибрати C_{n-2}^2, C_{n-4}^2, ..., C_2^2. Звичайно добуток цих величин не може бути реальною оцінкою складності тому що в цьому випадку має бути дійсним припущення про те, що у студентів чорні списки порожні і тоді ми можемо розбити їх на пари довільним чином і нам не треба перебирати багато комбінацій. Наявність студентів в чорних списках буде впливати на кількість елементів, із яких нам потрібно обрати наступну пару і при великих чорних списках ці множники можуть бути досить малими.

Чому ми взагалі розглядаємо дану проблему в контексті статті про атаки на систему дистанційної освіти? Тому що студенти можуть сформувати свої чорні списки таким чином, щоб зробити пошук можливого розбиття їх на групи максимально складним.

Якщо на деякому етапі після видалення пари вершин із графу у нас з'являється ізольована вершина, на потрібно повернутися на крок назад і спробувати видалити іншу пару вершин. Якщо жодна пара вершин не може бути видалена так, щоб не утворилася ізольована вершина, то потрібно повернутися ще на 1 крок назад.

Спроба поділу загальної навчальної групи на команди більшого розміру (3-5 осіб) означає, що окремі множники стають більшими ($C_n^3 > C_n^2$), але самих множників стає менше, так як на кожному кроці із графу видаляють не дві вершини, а 3,4 або 5.

І це лише перша проблема, пов'язана із роботою студентів у групі. Після того як ми розділити студентів на групи, нам потрібно гарантувати, що над проектом будуть працювати всі студенти в групі, а не хтось один. Тобто потрібно відслідковувати прогрес виконання завдання. І це ще не все. Навіть якщо студенти із групи вчасно завантажують свої проміжні звіти щодо виконаної роботи, ці звіти все ще можуть бути написані одним студентом, а потім розподілені між рештою студентів в групі і завантажені із відповідних облікових записів. Тобто в системі повинен бути модуль, який перевіряє "стиль" відповіді, щоб виявити випадки, коли всю роботу виконує один студент. Останньою проблемою, яку потрібно вирішити в цьому розділі є захист проекту. Якщо ми просто підготуємо список запитань, то один студент із групи може написати всі відповіді і роздати їх решті студентів у групі. З точки зору системи дистанційної освіти це виглядатиме так, ніби всі студенти, що працювали над проектом чудово розібралися із завданням і засвоїли матеріал. Цього допускати не можна. Під час захисту групового проекту студенти повинні отримувати запитання в режимі реального часу і одразу ж на них відповідати. І робити вони повинні це **одночасно,** щоб студент, який виконав увесь груповий проект не міг підказувати своїм колегам, бо в цей час він теж відповідає на питання. Таким чином система дистанційного навчання повинна одночасно видати студентам різні запитання щодо їх групового проекту, студенти повинні одночасно відповісти, дані відповіді потрібно записати, тому що викладач не може одночасно слухати відповіді 5 різних людей на 5 різних запитань, після чого викладач зможе прослухати ці відповіді послідовно і поставити відповідну оцінку. При такому режимі роботи студенти не зможуть використати інші засоби спілкування, які не контролюються системою дистанційного навчання (телефонний зв'язок, месенджери, електронну пошту)

Висновки. Система дистанційної освіти, як і будь-яка інша інформаційна система може стати ціллю для кібератаки. Але крім "класичних" кібератак, метою яких є використання ресурсів системи в інтересах порушника без огляду на призначення системи, існують інформаційні атаки, що пов'язані безпосередньо із функціями системи. Ці атаки найчастіше здійснюються користувачами даної системи для отримання неправомірної вигоди в контексті того бізнес-процесу, який дана система забезпечує. У випадку із системою дистанційної освіти, порушниками можуть бути як викладачі, так і студенти. Характерною особливістю атак цього виду є те, що їх "шкідливість" визначається не просто дією в системі, а зв'язком даної дії з іншими діями, що відбулися перед нею. Завантаження правильної відповіді і завантаження правильної відповіді, що є точною копією відповіді іншого студента, відбувається одними й тими ж засобами системи (в обох випадках студенти використовують програму клієнт для взаємодії із сервером, заповнюють одну й ту ж форму або прикріплюють файли подібного розміру та розширення). Але в

другому випадку відповідь є списаною і система дистанційної освіти повинна реагувати відповідно. Для реалізації системи дистанційної освіти необхідно впровадити наступні механізми/підсистеми:

1) розмежування доступу;

2) годинник/календар на сервері + мітки часу, обробник транзакцій, журнал, лог-файли;

3) перевірка метаданих (для завантажених файлів);

4) ідентифікація та автентифікація користувачів;

5) DRM, watermark, стеганографія, контроль запущених процесів на клієнті;

6) планувальник;

7) конвертація форматів файлів;

8) віртуалізація;

9) розподіл секретів (не через XOR), цифровий підпис, шифрування;

10) перекладач (якщо використовується декілька різних мов);

11) порівняння файлів або їх вмісту;

12) optical character recognition;

13) підтримка скриптів.

Список літератури

[1]. Diehl E. – Securing Digital Video: Techniques for DRM and Content Protection. 2012. DOI: 10.1007/978-3-642-17345-5_1, Springer-Verlag Heidelberg, ISBN-10: 3642434886, ISBN-13: 978-3642434884.

[2]. M. Asikuzzaman and M. R. Pickering, "An Overview of Digital Video Watermarking," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 9, pp. 2131-2153, Sept. 2018, doi: 10.1109/TCSVT.2017.2712162. keywords: {Watermarking; Streaming media; Motion pictures; Three-dimensional displays; Copyright protection; Internet; Robustness; Watermarking; robustness; geometric attack; depth-image-based rendering (DIBR); multi-view video};

[3]. Liu, Yunxia & Liu, Shuyang & Wang, Yonghao & Zhao, Hongguo & Liu, Si. (2018). Video Steganography: A Review. Neurocomputing. 335. 10.1016/j.neucom.2018.09.091.

[4]. Beimel, Amos. (2011). Secret-Sharing Schemes: A Survey. Coding and Cryptology. 11-46. 10.1007/978-3-642-20901-7_2.

[5]. Paul Heckel. 1978. A technique for isolating differences between files. Commun. ACM 21, 4 (April 1978), 264-268. https://doi.org/10.1145/359460.359467.

[6]. A. Zramdini and R. Ingold, "Optical font recognition using typographical features," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 877-882, Aug. 1998, doi: 10.1109/34.709616. keywords: {Optical character recognition software; Character recognition; Optical sensors; Bayesian methods; Text recognition; Image recognition; Text analysis; Feature extraction; Robustness; Optical design}.

[7]. Perwej, Dr. Yusuf & Hannan, Shaikh Abdul & Asif, Ali & Mane, Arjun. (2014). An Overview and Applications of Optical Character Recognition. International Journal of Advance Research in Science and Engineering (IJARSE). Vol. 3. Pages 261- 274.

[8]. John Barkley. 1997. Comparing simple role-based access control models and access control lists. In Proceedings of the second ACM workshop on Role-based access control (RBAC '97). Association for Computing Machinery, New York, NY, USA, 127–132. https://doi.org/10.1145/266741.266769.

УДК 004

Kireienko O. Attack scenarios on system of remote education

Abstract. *The development of both the violator model and threat model is required for protection of information system from potential harmful influence. Harmful influence can be caused by accident (by its legit users) or intentionally (by violators). Each model is an abstraction and the level of detail of this abstraction is determined by a few factors. One such factor is the object of protection. Systems of remote education are used by people in a wide range of ages – from 1st grade school students to adult university/college students, while the range of objects of attacks include educational materials and user accounts that contain personal data. The specifics of functioning of systems of remote education accounts for processing data in different formats and interactive, synchronous and asynchronous interaction of users who can have different roles in the system. One should also expect the abuse of main functions executed by legit users of the system and to watch over restriction of access to the system from external network.*

Keywords: *system of remote education, watermark, steganorgaphy, data format, feedback, sharing the secret, match search, optical character recognition (OCR).*

Кіреєнко Олександр Володимирович, асистент спеціальності «Кібербезпека» Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

Oleksandr Kireienko, assistant of the "CyberSecurity" specialty of the National Technical University of Ukraine "Ihor Sikorskyi Kyiv Polytechnic Institute".

Отримано *16 червня 2024 року*, затверджено редколегією *26 червня 2024 року*