МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

«Система захисту інформації підприємства на основі серії стандартів ISO/IEC 27000»

Завідувач
випускаючої кафедри **Довбиш А.С.**

Керівник роботи **Кальченко В. В.**

Студента групи КБ-71 **Шамонін К. Є.**

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерних наук

Затверджую_____
Зав. кафедрою Довбиш А.С.
"_____" _____2021 г.

ЗАВДАННЯ

до випускної роботи

Студента четвертого курсу, групи КБ-71 спеціальності "Кібербезпека" денної форми навчання Шамоніна Кіріла Євгеновича.

Тема: " Система захисту інформації підприємства на основі серії стандартів ISO/IEC 27000"

Затверджена наказом по СумДУ
№_____ от _____2021 г.

Зміст пояснювальної записки: 1) аналітичний огляд стандартів інформаційної безпеки; 2) опис процесу побудови системи інформаційного захисту; 3) розробка інформаційного й програмного забезпечення інтелектуальної системи; 4) аналіз результатів моделювання.

Дата видачі завдання "_____"_____2021 г.
Керівник випускної роботи _____ Кальченко В.В.
Завдання приняв до виконання _____ Шамонін К. Є.

РЕФЕРАТ

Записка: 57 стор., 8 рис., 1 додаток, 43 джерел.

Об’єкт дослідження – процес побудови системи захисту інформації підприємства на основі сімейства стандартів ISO/IEC 27000

Мета роботи – визначення процесу створення системи захисту інформації та сертифікація на основі сімейства стандартів ISO/IEC 27000 .

Методи дослідження – методи відвідання до стандартів інформаційної безпеки.

Результати – розроблено програмне забезпечення для забезпечення контролю доступу до інформаційних ресурсів підприємства. В процесі роботи вивчені та досліджені принципи побудови системи захисту підприємства, принципи сертифікації та стандартизації. Вивчені стандарти сімейства ISO/IEC 27000.

СТАНДАРТ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ, ІНФОРМАЦІЙНА БЕЗПЕКА, УПРАВЛІННЯ ДОСТУПОМ, СИСТЕМА УПРАВЛІННЯ ДОСТУПОМ, ISO/IEC 27000

ЗМІСТ

ВСТУП

Наразі у світі відбувається процес переходу до пост-індустріального суспільства. Також відбувається перехід від домінування постачальників у побудові бізнес процесів, до домінування споживача. Це означає, що підприємствам потрібно швидко змінювати вектор розвитку та адаптуватися до запитів ринку. Саме цьому підприємства надають перевагу переходу до ведення бізнес процесів використовуючи цифрові інформаційні засоби, які включають в себе електронний документообіг, використання комп'ютерного програмного забезпечення тощо [1]. Саме тому підприємства починають оперувати великими об'ємами даних, здебільшого в електронному вигляді.

Підприємства зацікавлені в підтримці безперервної роботи бізнес процесів та захисту інформації, яка забезпечує роботу компанії. Для цих цілей підприємства інвестують фінансові та часові ресурси у створення інформаційної інфраструктури та для її подальшого захисту. Мета захисту інформації – це захист важливих інформаційних активів підприємства, включаючи інформацію, комп'ютерне обладнання та програмне забезпечення. [2] Це необхідно для уникнення можливого порушення одного або декількох властивостей інформації: конфіденційності, цілісності, доступності. Поодинокі системи неефективні, тому спеціалісти використовують комплексні підходи для побудови комплексних систем захисту інформації для підприємств.

Метою цієї роботи є дослідження різних моделей побудови системи захисту інформації для підприємства, дослідження сучасних стандартів систем захисту інформації, розробка програмного забезпечення, що допоможе під час сертифікації підприємства відповідно до одного зі стандартів побудови системи управління інформаційно безпекою.

1. ЛІТЕРАТУРНИЙ ОГЛЯД

1.1. Передумова впровадження стандартів

Для максимальної ефективності дій направлених на забезпечення інформаційної безпеки підприємства необхідно поєднати відповідні дії у єдину комплексну систему захисту інформації. Система захисту інформації підприємства або система управління інформаційною безпекою (СУІБ) це набір політик, процедур, процесів, підходів, інформаційних систем та програмного забезпечення, які направлені на мінімізацію ризиків пов'язаних із безпекою даних в інформаційно-телекомунікаційних системах (ІТС). СУІБ підприємства описує загальні підходи не тільки до захисту інформації підприємства, але також описує дії направлені на забезпечення безперервної роботи бізнесу, процесу відновлення у разі інциденту інформаційної безпеки тощо.

Існують різні передумови для впровадженні СУІБ на підприємстві, але найбільш поширені є:

– необхідність у підвищенні загального рівня інформаційної безпеки підприємства.

– галузеві вимоги.

– тендерні або бізнес вимоги.

Необхідність підвищення рівня безпеки може бути обумовлено багатьма факторами, які можуть включати наступні фактори, але не обмежуються ними: економічне зростання підприємства, збільшення об'єму даних з обмеженим доступом, оновлення наявної або побудова нової інформаційної інфраструктури підприємств тощо. Також у кожній галузі є свої вимоги щодо захисту інформації, які регламентовані на законодавчому рівні або на рівні міжнародних стандартів. Тому для того, щоб отримати дозвіл для роботи у відповідній сфері, надавати послуги або продавати товари у певній країні – потрібно отримати відповідні дозволи для СУІБ підприємства. Наступною передумовою є тендерні або бізнес

вимоги, що обумовлюються необхідними вимогами для співпраці із підрядними організаціями або у ролі підрядної організації.

Під час впровадження СУІБ на підприємстві необхідно звернутися до порад спеціалістів або до стандартів у сфері інформаційної безпеки.

Основною перевагою порад спеціалістів є отримання сучасних даних щодо підходів, процесів, програмного забезпечення тощо. Це означає, що підприємство буде концентруватися лише на тих діях, які принесуть найбільшу перевагу для підприємства. Проте такий підхід актуальний лише для малих та дуже малих підприємств, які можуть швидко реагувати на зміни через те, що будувати нові та змінювати існуючи бізнес процеси легко і недорого. Також негативним фактором є підвищена вартість зовнішньої підтримки СУІБ через нечітко визначені вимоги та різну модель документації.

Стандарти, на відміну від неформальних вимог спеціалістів, надають формалізовані вимоги для побудови процесу управління безпекою інформації. Стандартизація це набір угод, яких повинні дотримуватися всі відповідні сторони галузі чи організації, щоб усі процеси, пов’язані зі створенням товару чи наданням послуг, виконувались у межах встановлених правил [3]. Стандартизація в рамках створення СУІБ переслідує такі ж цілі, що і у інших сферах бізнесу. Основні переваги стандартизації:

– чітке визначення вимог та характеристик суб’єктів та об’єктів СУІБ.

– забезпечення відповідності кожного об’єкта та суб’єкта СУІБ своєму призначенню.

– забезпечення раціонального використання ресурсів для впровадження та підтримки СУІБ.

– забезпечення більш чіткого розуміння моделі безпеки для менеджменту компанії.

– зменшення витрат на отримання дозвільних документів [4].

– підвищення впевненості у бізнес партнерів.

– маркетинг для споживачів послуг або товарів.

Чітке визначення вимог означає, що при побудові та підтримці СУІБ спеціалісти мають чіткий перелік необхідних політик, процесів, процедур тощо, які необхідно впровадити для того, щоб забезпечити необхідний рівень безпеки. Також це дозволяє спеціалістам із багатьох компаній спілкуватися єдиною термінологічною базою, що значно спрощує процес комунікації із іншими спеціалістами та робить процес супутньої підтримки більш дешевою. Також спеціалісти розуміють хід роботи та підтримки СУІБ у робочому стані.

Наступним фактором є отримання підтримки менеджменту організації через те, що формальні вимоги зрозумілі для менеджменту. Менеджмент має чіткі дані щодо характеру видатків для створення та підтримки СУІБ, тому охочіше підтримує спеціалістів та виділяє матеріальні ресурси.

Наступним фактором є зменшення витрат на отримання дозвільних документів. Це означає, що частина документації, процесів тощо були впроваджені та під час збору документів дозволяється посилатися на готові документи, процеси тощо. Також більшість міжнародних стандартів частково уніфікована, через що вартість впровадження кожного наступного стандарту зменшується.

Наступним фактором є підвищення довіри серед партнерів та споживачів послуг. Це досягається можливістю підприємства продемонструвати відповідальність, прозорість процесів та готовність до співпраці. Тим самим будуючи впевненість у організації всіх процесів, зокрема безпеки персональних даних.

1.2. Огляд існуючих стандартів

При побудові нової СУІБ підприємства на основі стандарту або подальшої стандартизації вже готової СУІБ необхідно обрати відповідний стандарт інформаційної безпеки. Існують 2 основні концепції стандартів. Перша концепція надає загальні вимоги до побудови СУІБ та загального захисту інформації. Прикладами є сімейства стандартів ISO, NIST тощо. Іншою

концепцією є чітко визначені галузеві стандарти. Прикладами є стандарти HIPAA, PCI DSS, FINRA, GDPR тощо. Також кожна держава впроваджує власні державні стандарти, наприклад в Україні побудована власна концепція захисту інформації – концепція комплексна система захисту інформації (КСЗІ). Через те, що стандартів багато та вони відрізняються функціями, перед вибором стандарту для побудови або сертифікації СУІБ потрібно зробити аналіз. Далі наводиться аналіз найрозповсюджених стандартів.

1.2.1. Стандарти сімейства NIST

Стандарти сімейства NIST були розроблені Національним інститутом стандартів у 2014 році як частина наказу президента США 13636 "Поліпшення кібербезпеки критичної інфраструктури", яке вимагало стандартизації системи безпеки критичної інфраструктури в США. Стандарти NIST визнаються багатьма країнами та організаціями, проте найбільш актуально для ведення бізнесу у США. [5] Стандарти сімейства NIST лягли у основі системи кібербезпеки (CSF).

Система складається з 3 основних частин: ядро стандарту (Framework Core), рівнів впровадження (Implementation Tiers), профілів (Profiles). [6]

Ядро системи це набір необхідних заходів та результатів роботи інформаційної безпеки, що згруповані у категорії з посиланням на додаткові інформаційні ресурси. Ядро було розроблено таким чином щоб бути легко зрозумілим. Також система надає єдиний простір для комунікації багатопрофільних команд, що значно полегшує роботу над СУІБ. Ядро включає в себе 5 основних категорій або функцій, а саме: ідентифікація, захист, виявлення, відповідь, відновлення. Далі категорії розбити на 23 під категорії, які покривають не тільки менеджмент безпеки, але і загальний підхід для ризик-менеджменту.

Ідентифікація стосується основи для побудови ефективної моделі інформаційної безпеки. Контроль, який описаний у цій категорії, здебільшого

сконцентрований на оцінці ризику, інвентаризації та створенні моделі обробки ризику. Кожне підприємство повинно визначити свої критерії оцінки ризику. зиків, інвентаризації IT-активів та створенні комплексної стратегії управління ризиками та задокументувати їх.

Захист стосується впровадження таких політик, процедур, процесів для належного захисту даних. Контроль, який описаний у цій категорії, здебільшого сконцентрований на навчанні співробітників, застосування політики контролю доступу, управління інформаційними активами, впровадження ПЗ для інформаційної безпеки.

Виявлення стосується впровадження механізмів аналізу, моніторингу та журналювання подій, тобто запис в хронологічному порядку подій, що відбуваються в системі. Здебільшого ця функція покривається спеціалізованим ПЗ, таким як SIEM.

Відповідь стосується впровадження плану забезпечення безперервної роботи бізнесу та плану відновлення після інциденту. Здебільшого це стосується підготовки планів відновлення після інцидентів інформаційної безпеки, що залучає різні рівні резервного відновлення, роботи менеджменту бізнесу та інших частин бізнесу.

Відновлення стосується процесу відновлення критичної інфраструктури, остаточного аналізу інциденту та внесення коригуючих дій.

Аналізуючи систему захисту можна прийти до висновку, що відповідне сімейство стандартів актуальне для компаній, що орієнтуються здебільшого на ринок США або територіально розташовані у США. Недоліком для впровадження системи захисту NIST в Україні є слабка підтримка спеціалістами, мала інформаційна база, складності у впровадженні та подальшій сертифікації через брак спеціалістів.

1.2.2. Стандарт PCI DSS

Наступним прикладом стандарту є стандарт PCI DSS (Payment Card Industry Data Security Standard). Він є галузевим стандартом для компаній, які займаються оперуванням банківських карток. Повна відповідність до цього стандарту надає можливість банківським компаніям та операторам платіжних систем оперувати інформацією про банківські рахунки, номерами карток, cvv кодами та іншою супутньою платіжною інформацією.

Стандарт бере свій початок з розвитку набору стандартів безпеки банківських карток. Згодом цей набор стандартів еволюціонував у стандарт PCI DSS під керівництвом Payment Card Industry Security Standards Council (PCI SSC), що був створений у 2006 році. [7] PCI SSC є незалежним органом, який підтримується основними операторами банківських карток, такими як Visa, MasterCard, American Express, Discover та JCB.

Стандарт надає вимоги до зберігання, обробки та передачі інформації щодо транзакції з використанням банківських карток. [8] Цей стандарт має на меті підвищення рівня безпеки інформації та зменшення ризику безпеки для відповідної інформації. Через те, що неможливо повністю захистись від кібер-атак та витоків даних, компанії, що відповідають вимогам стандарту, мають можливість мінімізувати можливий негативний вплив такого ризику.

Стандарт PCI DSS не надає єдиного сертифікату відповідності, проте вимагає щоб компанії відповідали вимогам. Для того, щоб компанія відповідала вимогам стандарту, компанія повинна пройти оцінювання за списком, що складається з 288 пунктів. Через те, що у кожної компанії є свій унікальний бізнес профіль, то частину пунктів необхідно пропустити. Компанії поділяються на 4 групи, базуючись на кількості транзакцій щорічно. Найменшим компаніям потрібно заповнити Self-assessment Questionnaire (SAQ), надати докази та надіслати SAQ на перевірку до вендора PCI SSC. Найбільші компанії повинні проходити аудит третьої сторони, що проводиться аудитором зі статусом Qualified Security Assessor.

Список відповідності PCI DSS включає в себе перевірку, що складається з таких основних розділів [9]:

1. Чи використовуються мережеві екрани для захисту даних?
2. Чи компанія відмовилась використання стандартних імен та паролів?
3. Чи дані про картки шифруються під час передачі?
4. Чи встановлений антивірусний захист на пристроях, на яких працюють оператори банківських карток?
5. Чи ПЗ та системи захищені?
6. Чи впроваджений контроль доступу до інформації?
7. Чи є у кожного об’єкта свій ідентифікатор для майбутнього аналізу та звітування7
8. Чи заборонений фізичний доступ до карток?
9. Чи відстежується доступ до інформації?
10. Чи проводяться регулярні тести безпеки?
11. Чи впроваджена політика безпеки на підприємстві?

Важливо відмітити, що PCI DSS є галузевим стандартом, який більше схожий на набір правил для підприємств та не несе законодавчої сили, на відміну від іншого стандарту EU GDPR (General Data Protection Regulation). [10] Це означає, що за недотримання вимог стандарту, відповідній компанії буде відмовлено у наданні послуг основними компаніями операторами карток або буде накладатися додаткова комісія з кожної транзакції. Щоденно може накладатися комісія від $10 до $100, проте підтримка відповідальності стандарту коштує від $1000 на рік.

Аналізуючи стандарт можна прийти до висновку, що відповідний стандарт актуальний для компаній, які оперують даними про банківські карки. Здебільшого це інтернет магазини, банки тощо. Для побудови системи захисту інформації для більшості підприємств цей стандарт може носити рекомендаційних характер та надавати додаткові підказки для побудови СУІБ.

1.2.3. Стандарт EU GDPR

Стандарт General Data Protection Regulation (GDPR) це є найсуворішим законом про конфіденційність та безпеку у світі. [10] GDPR був впроваджений у 2018 році та націлений на безпеку персональних даних громадян Європейського союзу. За словами ЄС, цей стандарт має на меті узгодити всі закони, стандарти, нормативні документи країн членів ЄС у галузі захисту персональних даних. [11] Також цей стандарт був розроблений для того, щоб змінити спосіб збору та обробки персональних даних. Цей стандарт поділяється на 99 частин.

GDPR відноситься до всіх організацій, що оперують на території країн ЄС, даними громадян ЄС або продають свої товари та послуги на території ЄС. Це фактично означає, що будь-яка міжнародна компанія або підприємство повинні відповідати вимогам GDPR. Цей стандарт поділяє учасників процесу обробки персональних даних на 2 категорії: обробники та контролери. Контролер – це особа, державний орган, відомство чи інший орган, який самостійно або спільно з іншими визначає цілі та засоби обробки персональних даних. Обробник – це особа, державний орган, агентство чи інший орган, який обробляє персональні дані від імені контролера. [12]

GDPR надає вимоги щодо обробки, зберігання, передачі персональних даних. Стандарт визначає персональними даними такі сутності як ім’я, адреса, фотографії, IP адреса, расова та генетична інформація, біометричні дані тощо, що може однозначно ідентифікувати особу. Також стандарт накладає більш суворі вимоги для захисту даних про расову та генетичну інформацію, про сексуальне життя або орієнтацію, політичні або релігійні погляди тощо. Компаніям та підприємствам необхідно безпечно обробляти дані, застосовуючи «відповідні технічні та організаційні заходи». [10]

GDPR надає 7 основних принципів. Принципи можна поєднати у такі основні групи: мінімізація даних, цілісність та конфіденційність даних, підзвітність. Ці принципи наслідують принципи, які були запропоновані у попередніх виданнях законів. Мінімізація даних означає, що організації повинні

збирати лише ті дані, якими користуються. Це необхідно для того щоб організації не збирали надлишкових персональних даних, які згодом можна використати для досягнення персональних інтересів бізнесу. GDPR не надає список кращих практик для забезпечення цілісності та конфіденційності даних, проте надає таку вимогу. Це необхідно для того, щоб користувачі відповідних ІТС були впевнені у надійності захисту їх персональних даних та для того, щоб відповідні чутливі дані не були опубліковані, потрапили до рук хакерів тощо, зберігаючи таємницю особистого життя. Підзвітність необхідна для того, щоб можна було відслідкувати хід даних у ІТС у ході перевірок, аудитори мали змогу перевірити хід зберігання та обробки персональних даних. [11]

Аналізуючи стандарт можна прийти до висновку, що відповідний стандарт актуальний для компаній, які оперують даними про громадян ЄС або зберігають чи оперують даними на території ЄС. Не зважаючи що на перший погляд ці вимоги відносяться до ЄС, фактично ці вимоги повинні бути виконаними практично у кожному підприємстві, організації компанії для запобігання накладання санкцій ЄС. Тому для побудови нових СУІБ потрібно враховувати вимоги GDPR у базовій архітектурі СУІБ.

1.2.4. Стандарти сімейства ISO 27000

Сімейство стандартів ISO / IEC 27000 було розроблено та опубліковано вперше у 2005 році підкомітетом спільного технічного комітету (ISO / IEC JTC SC27) Міжнародної організації зі стандартів (ISO). Цей стандарт надає модель, якою слід керуватись під час створення та експлуатації СУІБ. Ця модель базується на характеристиках, за якими експерти з інформаційної безпеки досягли консенсусу як єдиного бачення рівня технічного забезпечення безпеки інформації. Використовуючи цю модель, організації мають змогу розробити та впровадити СУІБ у компанії або на підприємстві, забезпечуючи безпеку інформаційних активів, фінансову інформацію, персональні дані, патенти тощо. [13] Модель також надає вимоги щодо незалежної сертифікації СУІБ

підприємства. Сімейство стандартів ISO/IEC 27000 є міжнародними стандартами, проте більшість держав впровадили національні стандарти, гармонізовані з міжнародними стандартами. Таким чином національною версією стандарту ISO/IEC 27001:2013 є ДСТУ ISO/IEC 27001:2015, ДСТУ ISO/IEC 27005:2019 є версією ISO/IEC 27005:2018 та інші. [14]

Сімейство стандартів ISO 27000 надає інформацію щодо:

- Вимог до створення та сертифікації СУІБ.
- Процесу створення, впровадження, підтримки та вдосконалення СУІБ.
- Галузевих настанови щодо СУІБ.
- Єдиної оцінки відповідності СУІБ.

Основними стандартами для побудови СУІБ є ISO/IEC 27001:2013 (далі ISO 27001) та ISO/IEC 27002:2013 (далі ISO 27002). Стандарт ISO 27001 являє собою специфікації для СУІБ. Цей стандарт є нейтральним та незалежним відносно до виробників та технологій, тому призначений до використання в усіх організаціях незалежно від типу, розміру, характеру та матеріально-технічної бази у будь-якій організації у світі. Цей стандарт являє собою систему управління, а не технологічними специфікаціями. У свою чергу стандарт ISO 27002 надає керівництво до впровадження технік, описаних у стандарті ISO 27001 у СУІБ організації. [15,16]

Стандарти сімейства ISO 27000 створені допомогти організаціям в управлінні ризиками інформаційної безпеки, внутрішніми загрозами безпеки даних. Під час зростання організації, архітектура ІТС стає все більш складною, а велика кількість технологічних застосунків відкриває все більше вразливостей, які не очевидні на перший погляд. [17]

Основні переваги стандартизації відповідно до ISO 27001 відповідають до загальних переваг сертифікації, проте стандартизація ISO 27001 є ключовою у побудові довірчих відносин із клієнтами та партнерами, через те що цей стандарт є загальновідомим, загальновживаним та є «необхідним стандартом» у уяві

потенційних партнерів. Також через те, що цей стандарт надає керівництво до загальної системи управління безпекою організації, загальний рівень обізнаності персоналу значно зростає та спеціалісти із безпеки не мають обмежень у інструментах для забезпечення відповідності вимогам стандарту. Стандарт ISO 27001 обумовлює створення політики безпеки, підходу до розподілу обов'язків, контролю безпеки території, процесу найму нових співробітників, контролю доступу тощо та не обумовлює чіткі інструменти для виконання вимог даючи можливість навіть невеликим підприємствам успішно пройти сертифікацію. [18]

Аналізуючи стандарт можна прийти до висновку, що відповідний стандарт є найбільш придатним для використання на підприємстві. Саме цьому цей стандарт був обраний як базис при побудові СУІБ для підприємства.

1.3. Постановка задачі

Метою даної роботи є створення плану розробки та впровадження СУІБ підприємства на основі сімейства стандартів ISO 27000, надання керівництва щодо кожного пункту плану та розробки моделі програмного забезпечення для контролю доступу до інформаційних активів підприємства.

Основними завданнями роботи є:

1. Описати план розробки та впровадження СУІБ на підприємстві.
2. Надати характеристику щодо кожного кроку плану, надати перелік необхідних для сертифікації документів, контролів.
3. Надати керівництво щодо процесів, які повинні бути впровадженні на підприємстві під час впровадження СУІБ.
4. Створити проект програмного забезпечення, що допоможе спеціалістам виконати один із процесів на підприємстві для впровадження СУІБ.

Задача проекту – створити модель програмного забезпечення для контролю доступу до локальних інформаційних ресурсів, що не підтримують протокол Lightweight Directory Access Protocol (LDAP) або включення в домен організації.

2. ПРОЦЕС ВПРОВАДЖЕННЯ СУІБ

2.1. Підготовка до впровадження

Побудова будь-якої системи управління інформаційною безпеки починається з отримання формального розпорядження, яке надається за підтримки керівництва. Рішення щодо впровадження приймається, коли штатні інструменти забезпечення захисту інформації більше не дозволяють бути впевненим у надійності захисту даних. Основними рушіями для впровадження системи є економічне зростання підприємства або оновлення інфраструктури.

Для кращого зрозуміння процесу впровадження СУІБ можна скористатися алгоритмом, зазначеним на рис 2.1.

Рисунок 2.1 – Алгоритм процесу впровадження СУІБ

2.1.1. Проектна команда

Спершу керівництво повинно визначити осіб, відповідальних за впровадження системи захисту інформації. [19] Основними дійовими особами є спеціалістами із інформаційної безпеки, спеціаліст із фізичної безпеки та проектний менеджер. Проте, якщо на підприємстві немає таких посад, тоді їх можуть замінити особи, виконуючі ці обов’язки.

Далі потрібно заручитися підтримкою менеджерів структурних підрозділів підприємства, які знайомі зі всіма внутрішнім процесами відповідних підрозділів. Такі менеджери повинні мати достатній об’єм повноважень, який

буде достатній для майбутнього впровадження нових процесів у відповідній зоні відповідальності. Ці співробітники будуть приймати участь у плануванні діяльності з впровадження та будуть відповідати за роботу у відповідній зоні відповідальності

Наступним кроком буде розподіл ролей та відповідальності. [20] Бізнес контекст організацій може відрізнятись один від одного, проте розподіл ролей повинен слідувати принципу RACI (відповідальний, підзвітний, консультований, поінформований). Невеликі підприємства можуть надавати схожі ролі на одного співробітника через брак компетентних людських ресурсів. Проте ролі співробітника не повинні бути конфліктуючими, тобто такими, що дозволяють виконувати завдання та робити приймання завдань.

Фінальним кроком збору проектної команди є створення формального документу, який визначає ролі, обов'язки та осіб, відповідальних за процес введення та підтримки СУІБ. Документ, який називається «Визначення ролей та відповідальності в галузі безпеки» регламентується п. 6.1.3 d стандарту ISO 27001. [21]

2.1.2. Створення проекту СУІБ

Впровадження СУІБ є проектом. PMBOK надає визначення проекту як «Тимчасові зусилля, спрямовані на отримання унікальної проектної послуги або результату». [22] Результатом успішного проекту є сертифікація підприємства згідно зі стандартом ISO 27001. Проект впровадження СУІБ на підприємстві зазвичай поділяється на три основні частини: аналіз та планування, впровадження процесів, акредитація.

Під час першої фази потрібно визначити ціль впровадження СУІБ, межі СІУБ, створити політику безпеки. Далі необхідно створити план роботи над проектом, провести необхідні аудити в межах проекту. Базуючись на аудитах потрібно створити план оцінки та обробки ризику, скласти положення про

застосовність. Тобто потрібно створити всі документи, що регламентовані п. 4.2.1 стандарту ISO 27001 [19].

Під час другої фази необхідно розробити та впровадити всі необхідні процеси та контролі, які були визначені під час фази аналізу та планування. Неможливо попередньо визначити список необхідних дій, що знадобляться під час впровадження без попереднього проведення аудитів, перевірок тощо.

Під час третьої фази відбувається кінцевий аудит сертифікованим аудитором. Після проведення аудиту та отримання сертифікації

2.2. Фаза аналізу та планування

2.2.1. Визначення меж впровадження СУІБ

Проектній команді із впровадження СІУБ потрібно визначити межі СУІБ. Визначення меж впровадження є критичним для майбутньої роботи із СУІБ. Визначення меж відбувається шляхом визначення частин ІТС підприємства, які повинні пройти сертифікацію. Межі повинні включати процеси, фізичні локації, сервіси, документи тощо, що повинно бути захищено та безпосередньо впливає на роботу ІТС підприємства. Це необхідно для того, щоб спеціалісти з впровадження та аудитори працювали тільки із певними частинами ІТС, що економить час та гроші на впровадження та оцінку СУІБ, бо аудитори дивляться на межі СУІБ, вплив елементів у та поза відповідними межами. Також це дозволяє продемонструвати партнерам або клієнтам які саме частини ІТС відповідають стандартам.

Організація повинна визначити межі виходячи із:

- Структура організації.
- Вимоги бізнесу.
- Місцезнаходження бізнесу
- Критичні процеси та продукти

Формальним документом, який описує межі є «Сфера застосування СУІБ», що регламентується п 4.3 стандарту ISO 27001.

2.2.2. Політика інформаційної безпеки

Політика інформаційної безпеки (далі політика) одна із важливих частин будь-якої СУІБ. Вимоги до створення політики на підприємстві регламентовані п. 5.1.1. стандарту ISO 27001. [22]

Ціль політики безпеки це надати керівництво щодо керування інформаційною безпекою у певній області з урахуванням цілей підприємства.

Політики поділяються на загальні та деталізовані. В загальних політиках потрібно вказати загальні підходи та правила. Загальні політики посилаються на більш детальні політики, які в свою чергу на меті мають реалізацію вказаних підходів. Загальні політики повинні бути зрозумілими до співробітників підприємства. Також відповідні загальні політики можуть бути передані до бізнес партнерів для визначення загальних правил співпраці, проте в такому випадку вся чутлива інформація повинна бути переміщеною до відповідних деталізованих політик. Більш деталізовані політики надають керівництво щодо певних аспектів СУІБ підприємства, процесів. Такі політики надають відповідальним особам більш чіткі вимоги для роботи. Приклад: політика використання паролів може бути частиною загальної політики безпеки підприємства, політика управління обліковими записами партнерів може бути частиною загальної політики управління обліковими записами у домені підприємства. Кожна політика повинна бути закріплена за власником та повинна переглядатися на щорічній основі. Це необхідно для підтримки даних в актуальному вигляді. Політика безпеки повинна бути об’єктом контролю версій, при будь-якому внесенні коректив в її зміст, необхідно створювати нову версію з переліком внесених змін. Також необхідно, щоб кожна версія політики була підписана відповідальною за неї особою та делегованою особою з керівництва.

Політика повинна містити достатньо інформації для розуміння процесів, але мінімально повинна містити:

- Опис зони використання, цілі політики та перелік визначень, які використовуються у політиці.
- Опис необхідності введення та ведення інформаційної безпеки на підприємстві.
- Перелік, опис загальних підходів та посилання на детальні інструкції або детальні політики для оцінки ризиків, менеджменту ризиків, додаткових контролів та інших процесів забезпечення безпеки інформації підприємства.
- Список відповідальних осіб, список та календар змін.
- Список виключень, які обробляє або не обробляє політика.

Політика повинна бути доступною для вивчення для працівників або зовнішніх партнерів.

При аудиті спершу перевіряються політики для отримання базового розуміння бізнес процесу та інтенцій керівництва. Далі відбувається перевірка на відповідність процесів до реальної операційної діяльності.

2.2.3. Аудит інформаційних активів

Для того, щоб проаналізувати наявні проблеми, отримати необхідну інформацію для підготовки наступних етапів, визначити які частини СУІБ ІС вже відповідають вимогам стандартів, а з якими потрібно проводити додаткову роботу, проводять аудити інформаційних активів. Немає сталого визначення терміну аудит, проте стаття [22] надає наступне визначення «Аудит – це процес збору та аналізу інформації про ІС для якісної або кількісної оцінки рівня її захищеності від атак зловмисників.»

Аудити поділяють на 3 типи: аудит першої сторони, другої сторони та третьої сторони.

Аудити першої сторони проводяться за допомогою виключно внутрішніх ресурсів. Вони дозволяють отримати знання про внутрішню організацію, дотримання вже наявних політик за відносно невеликі затрати часу та матеріальних ресурсів. Такі аудити повинні проводитися на постійній основі, яка відповідає цілям певного контексту.

Аудити другої сторони проводяться за рахунок зацікавлених сторін. Це можуть бути покупці товарів, послуг або зацікавлені особи від організацій партнерів.

Аудити третьої сторони проводяться неупередженими особами для того, що оцінити повноту документації, процесів та як вони виконуються. Такі аудити проводяться органами сертифікації або зовнішніми аудиторами за замовленням керівництва підприємства.

Первинний аудит підприємства необхідний для того, щоб зрозуміти якими даними оперує підприємство, які процеси вже встановлені та дозволяє оцінити сталість процесів.

Аудит інформаційних активів необхідний для того, щоб оцінити які дані наявні на підприємстві для встановлення певних рівнів критичності даних. [3]. Необхідно створити план внутрішнього аудиту, що наддасть внутрішнім аудиторам список дій, активів, ресурсів, процесів та процедур, які повинні бути перевірені. Через те, що внутрішні аудити зазвичай провдяться на циклічній основі потрібно створити єдиний план внутрішнього аудиту, який регламентований п. 9.2. стандарту ISO 27001, та заносити результати циклічного внутрішнього аудиту у журнал, що також регламентований відповідним пунктом.

Аудит повинен перевірити такі сутності:

- Електроні документи
- Паперові документи
- Електрона пошта
- Інформаційні системи та бази даних

Засоби зберігання даних (жорсткі диски, флеш-накопичувачі тощо)

Стандарт ISO 27001 не регламентує класифікацію даних, через різний контекст бізнесу, проте надає пропозицію щодо класифікації яка базується на шкоді до підприємства.

Запропонована класифікація даних:

Публічна – інформація, яка вільно поширюється.

Для внутрішнього використання – інформація з частково обмеженим доступом, яка використовується у повсякденних задачах. Порушення Конфіденційності Цілісності Доступності (КЦД) Може призвести до зовсім невеликої шкоди.

Обмежений доступ – інформація, порушення КЦД якої призведе до помірної шкоди до підприємства у певному регіоні. Може призвести до втрати конкурентоспроможності у відповідному регіоні.

Конфіденційна інформація – інформація, порушення КЦД якої призведе до катастрофічної шкоди для підприємства на світовому рівні. Може призвести до банкрутства компанії.

Після аудиту інформаційних ресурсів – відповідні ресурси повинні бути промарковані відповідно до їх класифікації. Пакети сучасних офісних програм мають вбудований функціонал для маркування електронних документів. Програми типу DLP можуть аналізувати маркування документів під час передачі або доступу до даних, для запобігання витоку інформації з обмеженим доступом. Фізичні документи можуть маркуватись у верхньому кутку відповідно до класифікації.

Наприкінці цього етапу необхідно створити наступні документи:

- «Реєстр активів», який регламентується п. 8.1.1.
- «Допустиме використання активів», який регламентується п. 8.1.3.

2.2.4. Оцінка ризиків

Оцінка ризиків лежить в основі будь-якого проекту з побудови СУІБ будь-якої організації. Цей процес є надзвичайно важливими для забезпечення того, що СУІБ всебічно та належним чином розглядає загрози. Оцінка ризиків в інформаційних системах - це процес ідентифікації та оцінка ризиків в області інформаційної безпеки. Цей процес дозволяє організаціям оцінювати та керувати інцидентами, які потенційно можуть завдати шкоди конфіденційним даним. Проводячи оцінку ризиків, організації дізнаються, наскільки уразлива інформаційна інфраструктура та активи, планують необхідні дії для пом'якшення наслідків.

На сьогодні актуальними є системи і методології управління ризиками, такі як стандарти NIST, Facilitated Risk Analysis Process (FRAP), Operationally Critical Threat, Asset and Vulnerability Evaluation (OCTAVE) та ISO/IEC 27005. Вони є широко використовуваними галузевими стандартами. Серед них NIST SP 800-30 / 37, ISO / IEC 27005 та OCTAVE – це загальні моделі оцінки ризиків в області інформаційної безпеки з урахуванням специфіки IT. Такі моделі високого рівня визначають лише загальні підходи або керівництво дій з оцінки ризиків, проте є неактуальними як методології, що базуються на конкретних метриках оцінки ризику. Ці методології недостатньо концентруються на оцінці ризику, в них не вистачає конкретних метрик ризику і автоматичних методів розрахунку ризиків, проте в основному зосереджені на чітко визначеному процесі аудиту в галузі інформаційної безпеки.

Для оцінки ризику системи, в першу чергу, необхідно визначити метрики за якими будуть проводитися вимірювання. В IT-галузі метрики безпеки щодо незрілі. У порівнянні із іншими областями, такими як фінансові ризики, немає всеосяжного набору метрик. Таким чином, існує необхідність у визначенні нових метрик, специфічних для інформаційних систем, обов’язково враховуючи сучасні тенденції розвитку технологій. Такі моделі розвитку мереж, як Bring your own device (BYOD), вимагають більших зусиль для комплексної оцінки ризику.

Однак, щоб бути корисною, хороша метрика повинна бути відтворюваною (при послідовних вимірах), дешевої для збору, враховувати контекст і мати одиницю виміру.

Після визначення метрик, як одиниць виміру для оцінки ризику, наступним етапом є опис і виявлення вразливостей. Для додаткового збору даних про характеристики, вплив і різні типи ризиків широко використовується відкритий фреймворк під назвою CVSS. [23]

Необхідно провести аналіз ризику який включає в себе:

- Інформаційні системи.
- Технічну інфраструктуру.
- Фізичну безпеку.
- Роботу кожного відділу.
- Роботу бізнесу.

Необхідно проаналізувати ризик за впливом на конфіденційність, цілісність та доступність інформації. Також потрібно проаналізувати ризик, який може відбутися при порушенні конфіденційності та/або цілісності та/або доступності.

Хід оцінки ризику, критерії тощо потрібно відобразити в політиці оцінки інформаційного ризику. Політика регламентується пунктом 6.1.2 стандарту ISO 27001.

2.2.5. План обробки ризику

Після аналізу ризику необхідно розробити процес обробки ризику. Цей план висвітлює кроки, які будуть зроблені для мінімізації ризику. Метою обробки ризику є контроль ризиків, що були знайдені під час етапу оцінки ризиків. Це означає зменшення ризику за рахунок зменшення ймовірності інциденту та/або зменшення впливу на активи, такі як бізнес процеси, інформація тощо.

Стандарти ISO надають наступний список можливих варіантів обробки ризику:

- Зміна ризику.
- Запобігання ризику.
- Розподіл ризику.
- Збереження ризику.

Зміна ризику є найбільш популярним та найбільш вживаним способом мінімізації ризику Зміна ризику досягається шляхом зміни частоти появи ризику та/або зміни результату ризику. Зміна ризику може досягатись використанням додаткових інструментів забезпечення безпеки, впровадженням політик або зміною бізнес процесів. Наприклад, впровадження політики використання паролів дозволяє зменшити ризик неавторизованого доступу у систему, впровадження політики використання сервісних облікових записів у системі дозволяє зменшити ризик неавторизованого доступу до конфіденційної інформації. [25]

Запобігання ризику. Існують безліч способів запобігання ризику, але найбільш ефективним є відмова від активності, що провокує ризик, наприклад відмова від використання даного типу програмного забезпечення, відмова від цього типу бізнес процесів. Проте не завжди цей спосіб можливо використовувати. Наступний спосіб мінімізації ризику є впровадження додаткових інструментів забезпечення безпеки, які допоможуть у запобіганні ризику. Наприклад, встановлення паркану, впровадження системи пропусків та запрошення охорони для захисту фізичної безпеки допоможе запобігти ризику проникнення сторонніх пересічних осіб на територію підприємства. Цей тип зменшення ризику є найбільш ефективним, але потребує найбільших фінансових ресурсів або зменшує задоволеність працівниками у робочому процесі, що може призвести до того, що співробітники будуть намагатися обходити створені обмеження.

Розподіл ризику – процес розподілу ризиків із третьою стороною. Існує два основних способи розподілу ризиків: страхування та підписання контрактів з підрядними компаніями. Страхування актуальне у випадку надзвичайної події, наприклад страхування від пожежі. Інший спосіб – контракти із профільними підрядними компаніями, наприклад використання охоронного підприємства для забезпечення фізично безпеки, оренда комерційного Security Operations Center для підвищення швидкості реагування на інциденти. Варіант розподілу ризику працює лише у парі із попередніми пунктами: зміна ризику, запобігання ризику.

Прийняття ризику – означає, що менеджмент підприємства згоден зі всіма наслідками, якщо ризик буде приведений у дію.

Правильний підхід до оброки ризику означає , що обрані всі необхідні засоби контролю та не включені зайві елементи. Обрані засоби контролю покривають всі ризики. Неправильний підхід означає, що були обрані неефективні засоби контролю, мають неоправдану вартість впровадження або користувачі систем не мають можливості виконувати свої робочі обов’язки.

Щоб гарантувати, що обробка ризиків інформаційної безпеки є ефективною та ефективною, тому важливо бути готовим продемонструвати зв'язок від необхідних засобів контролю до результатів оцінки ризиків та процесів обробки ризиків. Часто для реалізації зазначеного трактування ризику безпеки знань необхідно використовувати декілька засобів контролю, якщо обраний вибір варіації результатів конкретної події, може знадобитися контроль, щоб здійснювати оперативне виявлення події також як засоби контролю відповісти на подію та відновити її.

Визначаючи контроль, організація повинна також враховувати засоби контролю, необхідні для послуг від сторонніх постачальників, наприклад додатки, процеси та функції. Як правило, ці засоби контролю передбачаються шляхом введення вимог щодо захисту інформації в рамках угод з цими постачальниками, включаючи способи отримання інформації, близькі до того, наскільки ці вимоги виконуються (наприклад, право аудиту). Також можуть бути

ситуації, коли організація бажає розробити та описати детальний контроль як частину власної СУІБ, хоча контроль здійснюється сторонніми постачальниками. Незалежно від підходу, організація завжди повинна враховувати засоби контролю, необхідні своїм постачальникам, при визначенні засобів контролю для своїх СМІБ.

Після аналізу ризику необхідно заповнити таблицю Statement of applicability (SoA). Ця таблиця складається зі 114 пунктів контролю, опис дій які будуть проведені для мінімізації наявних у таблиці ризиків. Потрібно описати дії які були впроваджені для мінімізації наявного ризику, докази впровадження, причини для прийняття ризику. Таблиця SoA це один із перших документів, які перевіряються аудиторами. Необхідно однозначно вказувати всі контролі, так як аудитори можуть трактувати відповідний контроль подвійно. Це може бути одним із перших аргументів у відмові під час аудиту.

Стандарт ISO 27001 не надає єдиної структури плану обробки ризику, проте для кожного ризику потрібно покрити наступні пункти:

- Найменування ризику.
- Обраний спосіб обробки.
- Необхідні дії для впровадження.
- Статус впровадження.
- Відповідальний за ризик.
- Залишковий ризик.

Якщо потрібні будь-які дії для зменшення ризику, тоді потрібно запланувати відповідні дії під час другої фази впровадження стандарту. План обробки ризику зазвичай групує ризики по групам, надає інформацію про строки впровадження, відповідальних за ризик.

Коли були створені SoA та план обробки ризику – вони повинні бути затверджені у менеджменту підприємства, відповідальних за ризик. Менеджмент підприємства повинен бути згоден із витратами ресурсів та часу, які будуть витрачені для мінімізації наявного ризику. При завищених витратах – потрібно

створити новий план обробки, обрати інші інструменти обробки, отримати дозвіл на прийняття більшої кількості ризику.

Створення SoA регламентується пунктом 6.1.3 стандарту, створення плану обробки ризику регламентується пунктом 6.1.2 стандарту.

Після закінчення процесу менеджменту ризиків потрібно створити документацію, яка регламентується п. 8.2. стандарту ISO 27001 та у майбутньому наддасть інформацію спеціалістам по необхідним діям.

2.3. Фаза впровадження процесів

Під час фази впровадження процесів проектна команда разом із менеджментом підприємства повинні впровадити процедури та процеси, які дозволять мінімізувати ризики, які були ідентифіковані під час аудиту інформаційної безпеки, та впровадити додаткові процеси, що регламентовані стандартом. Під час фази впровадження процесів, проектний менеджер разом зі спеціалістами з впровадження СУІБ повинні спершу переглянути повинні переглянути таблицю ризиків та створити проектний план фази впровадження процесів.

Далі наведений перелік процедур, документів та процедур, що повинні бути впроваджені на підприємстві, що регламентовані стандартом ISO 27001. Також наводяться додатковий опис процедур тощо, що може допомоги під час створення СУІБ. Цей перелік не є кінцевим, так як бізнес контекст будь-якого підприємства відрізняється один від одного та для кожного відповідного підприємства потрібно впроваджувати свій унікальний набір процесів, але не менший аніж наведений далі.

2.3.1. Процедура контролю документації

Стандарт ISO 27001 надає вимоги до створення процедури контролю документації. Мета цієї процедури – забезпечити процес створення, оновлення,

ведення документації СУІБ. Це необхідно для того, щоб бути впевненим, що документація завжди знаходиться в актуальному стані.

Стандарт регламентує, що процедура контрою документації має містити наступні вимоги:

– Документація повинна мати чіткий заголовок та зрозумілий опис для чого використовується відповідний документ.
– Докази того, що документ переглядається та затверджується на циклічній основі.

Найкращим способом досягнення відповідності до цих вимог є створення електронної бібліотеки, де буде зберігатися вся необхідна документація. Доступ до цієї бібліотеки можна надати до всіх співробітників, обмеживши доступ для відповідних співробітників лише до документів, необхідних для виконання їх службових обов'язків. Також позитивним фактором впровадження бібліотеки є використання посилань, що зменшує кількість дубльованих даних, зменшуючи трудовитрати персоналу та збільшуючи [24]. Створення процедури контролю документації регламентується п. 7.5 стандарту ISO 27001.

Також у процедурі потрібно описати процедуру зберігання інформації. Необхідно описати загальний підхід до зберігання інформації, описати вимоги до зберігання та видалення інформації. Необхідно періодично видаляти застарілу інформацію, яка може становити загрозу ризику, також потрібно видаляти будь-яку персональну інформацію з документації, згідно вимогам стандарту GDPR.

2.3.2. Процедура безпечного оточення

Відповідна процедура має на меті впровадження необхідних дій, щоб запобігти неавторизований фізичний доступ до периметру підприємства та фізично захистити обладнання.

Ця процедура повинна містити вимоги та процес забезпечення безпеки. Необхідно визначити периметр безпеки на підприємстві та поділити підприємство на декілька зон безпеки, до яких мають змогу потрапити тільки

відповідні особи. Доступ до зон потрібно обмежувати за допомогою систем електронних чи фізичних ключів. Також потрібно використовувати системи слідкування, наприклад камери, журнал подій та доступу.

Для того, щоб потрапити усередину зовнішнього периметру співробітник або відвідувач повинні пройти необхідний контроль відвідувачів. Це може бути охоронний пост, турнікет, рамки метал детектору. Це необхідно для журналювання доступу до внутрішнього периметру та для того, щоб впевнитись у зменшенні ризику. Також необхідно, щоб відвідувачі були чітко визначені, наприклад спеціальний бейдж, та вони завжди знаходились поряд із відповідальним співробітником. Аудитор очікує побачити наявність відповідного контролю, а також регулярне тестування та моніторинг. [26]

Окрім підвищення рівню безпеки внутрішнього периметру, також потрібно підвищити рівень безпеки зон завантаження на підприємстві. Це необхідно через те, що до цих зон мають доступ неавторизовані особи, наприклад водії чи оператори спеціальної техніки. Відповідні зони повинні бути повністю або частково ізольовані від фізичного доступу усередину та від засобів, що обробляють інформацію. В таких зонах потрібно встановлювати додаткову охорону, камери. Співробітники в таких зонах мають використовувати персональні переносні термінали радше ніж стаціонарні термінали.

Усі дії для забезпечення фізичної безпеки повинні бути формалізовані та міститись у відповідній процедурі. Також необхідно мати формальний план місцевості. Зовнішній аудитор повинен переконатись, що жодна неавторизована особа не зможе отримати доступ до конфіденційної інформації.

2.3.3. Процедура утилізації та знищення

Стандарт регламентує створення процедури утилізації та знищення, що передбачає виконання обох пунктів 11.1.2 та 8.3.2. Ці пункти вимагають, щоб все обладнання, що могло містити або містить будь-які дані, було перевірено на предмет того, що будь-які дані були видалені або перезаписані перед тим як

повторно використовувати чи знищувати відповідне обладнання. Також необхідно щоб будь-які дані були знищені, якщо більше не використовуються. [25]

Ця процедура повинна містити вимоги, що необхідно зробити з обладнанням. Найкращий спосіб для утилізації обладнання – утилізація обладнання та засобів, які можуть зберігати інформацію окремо. Постійні запам’ятовуючі пристрої повинні бути повністю перезаписані 3 рази та/або піддатися впливу, що повністю унеможливить зчитування інформації: дія магнітного поля, механічна дія.

2.3.4. Процес забезпечення безперервної роботи бізнесу

Процедура обробки інциденту йде поруч із процесом забезпечення безперервної роботи бізнесу. Вимагається стандартом у п. 17.1, щоб на підприємстві був процес безперервної роботи бізнесу, а вимоги до інформаційної безпеки були частиною цього процесу. [31]

Ця процедура повинна надавати вимоги до швидкості відновлення критичних компонентів та процесів, які можуть блокувати виконання підприємством своєї виробничої функції. Під час аудиту були визначені критичні процеси, які можуть призвести до простоїв, відповідні процеси повинні бути відновлені у найкоротший термін та бути захищені краще всього. Відповідні процеси повинні бути класифіковані використовуючи інструмент аналізу впливу на бізнес, створення відповідного інструменту регламентує п. 17.1.1. Відповідний інструмент повинен використовуватись для оцінки можливого впливу на бізнес процеси. Вкрай важливо будувати план відновлення роботи базуючись на цьому інструменті, через те, що він надає інформацію щодо впливу на роботу бізнесу.

Для пришвидшення відновлення краще всього розробити процедуру резервного копіювання. У відповідній процедури потрібно визначити вимоги до створення, зберігання, підтримки резервних копій та процес відновлення з

резервної копії. Загалом вимоги до резервного копіювання повністю залежать від вимог плану забезпечення безперервної роботи бізнесу. Створення процедури регламентовано п. 12.3.

2.3.5. Політика безпеки відносно постачальників та партнерів

Жодне підприємство не може побудувати ефективні бізнес процеси без партнерів та постачальників товарів чи послуг. Підприємства можуть купувати первинну продукцію для переробки у товари для майбутніх покупців також підприємства можуть користуватись послугами логістичних партнерів для доставки готової продукції кінцевим споживачам. Саме через те, що між партнерами йде обмін інформації, що може містити чутливі дані, потрібно бути впевненим, що відповідна інформація буде захищена.

Політика безпеки повинна визначати вимоги до безпеки інформації для партнерів при роботі із підприємством. Проте, деякі постачальники послуг та товарів можуть бути значно більшими організаціями аніж підприємство, на якому ведуться роботи із впровадження СУІБ. Також менеджмент підприємства зацікавлений в максимізації прибутку та не може втрачати прибуток через неоправдані заборони. Саме тому політика безпеки відносно постачальників повинна містити вимоги щодо перебігу, обробки та зберігання даних. Також в цій політиці потрібно описати які вимоги бажано, щоб були виконані. Наприклад, обов’язково, щоб дані передавалися безпечним шляхом у зашифрованому вигляді, алгоритми шифрування повинні бути не менш ефективними ніж AES-256. Або, бажано, щоб у партнера була сертифікована СУІБ за державним або міжнародним стандартом.

Така політика повинна передаватись постачальникам перед початком роботи над контрактом для того, щоб постачальник мав змогу підготуватись до початку співпраці. Саме цьому бажано впровадити сертифіковану СУІБ на підприємстві для того, щоб відповідати вимогам більшої кількості

постачальників. Вимоги до політики регламентовані п. 15.1 стандарту ISO 27001. [28]

2.3.6. Процедура обробки інцидентів

Під час діяльності підприємства, можуть відбуватися різні інциденти. Міжнародні нормативні акти надають власне визначення інциденту. За їх баченням, інцидент це поодинока, непередбачувана і небажана подія, що може призвести до впливу на бізнес процеси в організації, тобто порушити, скомпрометувати, знизити рівень безпеки тощо. [29] На підприємстві інцидентами можна класифікувати залишені документи без нагляду або dos-атака.

Для того, щоб створити ефективну процедуру обробки інцидентів необхідно провести дійсно багато роботи, бо п. 16.1 стандарту надає основною вимогою, щоб відповідна процедура дозволяла швидко відновити роботу підприємства після інциденту. Для цього необхідно поділити інциденти на класи, описати відповідні класи та зазначити відповідну класифікацію у формальній документації. Також відповідна документація повинна містити опис ієрархії подій за рівнем їх потенційної шкоди. Це необхідно для опису коригуючих дій. Під час опису коригуючих дій необхідно зауважити частоту інцидентів, їх потенційну шкоду, метрики. В цьому випадку краще за все використовувати інформацію із відкритих джерел або від бізнес партнерів. В такому випадку спеціалісти отримають більш релевантну інформацію.

Також процедура повинна надавати керівництво дій у разі інциденту та для відновлення. Потрібно описати процес реагування на інциденти. В цьому процесі повинно описати яким чином та яка особа повинна повідомити про інцидент. Також потрібно вести облік інцидентів безпеки. В цьому випадку найкращим інструментом буде використання SIEM системи від одного з багатьох вендорів. SIEM це системи моніторингу подій та інцидентів безпеки, відповідні системи можуть повідомляти про можливі інциденти та заносити дані про інцидент до

внутрішнього журналу. Також потрібно описати процес інформування співробітників про можливий інцидент, як він вплине на роботу та які коригуючі дії повинні бути зроблені загальним персоналом підприємства, навіть якщо цей персонал не має достатньої компетенції у сфері інформаційної безпеки. Після того, як інцидент був вичерпаний, робота була відновлена, потрібно зробити висновки, тому процедура також повинна мати керівництво щодо збереження інформації про досвід що був отриманий в процесі вирішення, також потрібно включити вимогу щодо додавання доказів що відповідний інцидент не повториться знову. [30]

2.3.7. Процес контролю змін

Кожне оновлення процесу, ПЗ тощо повинно проходити формальний процес контролю змін. Цей процес необхідний для того, щоб мінімізувати можливий ризик до бізнес процесів та підвищити обізнаність відповідального персоналу. На підприємстві потрібно запровадити комісію зі змін, яка буде допомагати із проведенням аналізу змін. Ця комісія повинна вивчити зміни, що планує впровадити співробітник та прийти до висновку чи безпечні ці зміни чи ні, та надати список дій, які потрібно виконати співробітнику для того, щоб бути впевненим, що ці дії не підвищать певний рівень ризику для безперервної роботи бізнесу, викладений у плані безперервної роботи бізнесу.

Кожне оновлення та зміна повинні проходити через формальний процес оцінки ризику для бізнесу, тестуватись у тестовому середовищі та перевірятись відповідальними співробітниками. Це необхідно для того, щоб бути впевненим у тому, що оновлення будь-якого ПЗ не створить шкоду та простої підприємства. Також бажано використовувати тільки нове програмне забезпечення, що не старше ніж 2 версії, що дозволяє отримувати актуальну технічну підтримку та в такому ПЗ менше можливих вразливостей.

Потрібно описати процес контролю за змінами, визначити відповідальних осіб та створити план дій для замовника відповідного процесу. Цей процес регламентований п. 12.1 та 14.1 стандарту ISO 27001.

2.3.8. Процес безпечної розробки програмного забезпечення

Якщо підприємство використовує програмне забезпечення власного виробництва, то підприємство повинно приділяти увагу до процесу розробки відповідного ПЗ. Відповідно до визначення, безпечна розробка програмного забезпечення це застосування практик безпеки при розробці внутрішніх проектів з розробки ПЗ. [33] Для цього варто враховувати безліч факторів навколишнього середовища та людського фактору.

Потрібно описати загальні правила до розробки. Це вимагається п. 14 стандарту. Стандарт на надає чітких вимог, проте розробники повинні використовувати архітектурні прийоми, кращі підходи та практики, які допоможуть знизити загальний рівень ризику при розробці. Принципи безпечної розробки повинні бути прийняті до уваги на кожному етапі розробки ПЗ. Потрібно бути впевненим, що інформація залишиться захищеною під час обробки, зберігання та передачі.

Всі правила повинні бути викладені формально та бути доступними для внутрішніх розробників. Не завжди відповідні правила працюють, так як лише небагато підприємств займаються розробкою ПЗ.

2.3.9. Журнал користувацьких дій, виключень, подій безпеки

Для того, щоб проводити розслідування інцидентів інформаційної безпеки потрібно вести журнал дій користувачів, наявних виключень у роботі СУІБ та процесах, будь-яких подій безпеки. Цей процес потребує багато технічних ресурсів на реалізацію, через те, що потрібно вести опис дій цілого підприємства. Цей процес можна розділити на 2 основні частини: журнал користувацьких дій та журнал подій безпеки.

Журнал користувацьких дій це підхід до зберігання інформації, про дії, що були зроблені користувачем. [34] Бізнес процеси кожного підприємства відрізняються, тому спеціалістам потрібно використовувати різні методи збору, опису та зберігання дій користувачів, також спеціалісти повинні збирати лише ті дані, які потрібні для аналізу та які не підпадають під GDPR. Для цього можна використовувати ПЗ, що може бути інтегроване з багатьма сучасними API, мережевими платформами тощо. Непоганим інструментом є SIEM системи. Такі системи збирають записи користувацької активності, подій, статистики та метрик. Сучасні SIEM системи від провідних вендорів згідно з квадраном Гартнера можуть бути інтегровані в сотні сервісів, мати можливість у режимі реального часу збирати та аналізувати дані для того, щоб спеціалісти мали змогу реагувати на інциденти значно швидше. Також спеціалісти з розробки можуть створити програмні модулі, які будуть покривати всі необхідні області у відповідного підприємства. Проте SIEM системи вимагають кваліфікованих кадрів. Тому потрібно описати у відповідній процедурі, що вимагається п. 12.4, як саме спеціалістам ІБ потрібно збирати та аналізувати відповідні дані.

Створення журналу подій безпеки може бути значно важчим. Потрібно збирати статистичні дані зі всіх джерел. Потрібно збирати статистичні дані з виробничих ліній, засобів контролю фізичного доступу тощо та зберігати на захищених носіях. Сучасні підприємства записують журнал дій на магнітну плівку, що може зберігати великий об’єм даних для подальших розслідувань. Відповідні процеси також повинні бути описані. В описі цих процесів потрібно приділяти увагу до виключень та до прийнятих ризиків, знайдених під час аудиту.

2.3.10. Політика передачі даних

Неможливо уявити ефективне підприємство, яку не передає, отримує та обробляє дані. На підприємстві дані можуть поступати у електронному вигляді, на фізичних носіях. Стандарт ISO 27001 регламентує забезпечення безпечної

передачі інформації усереднені організації та з зовнішніми сутностями, тобто партнерами, організаціями, підрядниками тощо.[35]

Ця політика повинна бути наслідником процедури оцінки та класифікації даних, тому що до передачі різних типів інформації можуть накладатись різні вимоги щодо передачі. Відповідальний за впровадження персонал повинен розробити вимоги для передачі даних. Наприклад, для передачі критичних даних потрібно використовувати шифрування, а для передачі загальної бізнес інформації усереднені підприємства можна не використовувати шифрування.

Взагалі механізми безпечної передачі даних можуть змінюватись та відрізнятись один від одного. Якщо підприємство зберігає інформацію у зовнішньому файловому сховищі, тоді підприємство повинно створити умови для безпечної передачі інформації до віддаленого сховища використовуючи VPN тощо. Якщо підприємство використовує локальний метод збереження даних, тоді немає необхідності ускладнювати архітектуру.

Вимоги щодо безпечної передачі інформації повинні бути включені у політику безпеки відносно постачальників та партнерів. Також створення політики безпечної передачі даних регламентовані п. 13.1 стандарту ISO 27001.

2.3.11. Процедура чистого столу та екрану.

Відповідна процедура повинна регламентувати співробітникам вимоги до зберігання фізичних документів на столі або будь-якої інформації на екрані. Через те, що документ, який залишився без уваги містить більшу загрозу для підприємства, усі співробітники повинні слідкувати за документами, нотатками, чернетками, які були залишені без нагляду. Співробітники повинні пристібати їх обладнання до монолітних конструкцій за допомогою замку, зберігати будь-які записи у шухлядках із замком. Також потрібно використовувати персональні принтери або мережеві принтери лише із можливістю друку коли співробітник поруч.

Іншими вимогами є вимоги до сесії роботи, тобто часу від автентифікації до періоду завершення роботи. Потрібно визначити вимоги до завершення сесії при не активності користувача. Це необхідно для того, щоб зловмисник не отримав доступ до обладнання, яке було помилково залишено користувачем. Зазвичай період не активності складає від 10 до 20 хвилин до періоду роз'єднання. [27]

Відповідні вимоги визначені п 11.2.9 стандарту.

2.3.12. Процес контролю доступу

Будь-яка організація повинна впровадити процес контролю доступу, що базується на принципі мінімальної достатності. Це дозволить бути впевненим, що співробітник не зможе отримати неавторизований доступ до ресурсів, що можуть містити чутливу інформацію. Управління доступом є однією з ключових частин впровадження ефективної СУІБ. Система управління доступом використовується для управління і моніторингу доступу користувачів до файлів, систем і сервісів, щоб допомогти захистити організації від несанкціонованого доступу до інформаційних активів через порушення одного із властивостей інформації – конфіденційності. Створення політики контролю доступу регламентується п. 9.1 стандарту ISO 27001.

Управління доступом - це контроль користувацького доступу, що включає в себе відстеження і зміна повноважень при необхідності. За замовчуванням повинен бути контекст, що заборонено все, окрім дозволеного. Це одна із найбільш простих сфер для реалізації, через те, що є багато сервісів контролю доступу, які мають можливість інтеграції із системами підприємства, проте персоналу, що займається питаннями доступу потрібно розподілити повноваження так, щоб не порушувати бізнес процеси та не заплутувати співробітників. Через те, що будуть введено занадто суворі обмеження, користувачі будуть сидіти та чекати того, щоб банально отримати доступ до

ресурсу, а власники ресурсів будуть увесь час переглядати доступ до ресурсів. [37]

Найкращою моделлю контролю доступу є модель доступу основаною на ролях (RBAC). У кожному ресурсі є права доступу. Це повноваження на перегляд чи зміну певних параметрів. Зазвичай такі права об'єднуються у ролі, що значно простіше відслідковувати та контролювати. Також потрібно розділяти ролі на критичні та некритичні за розподілом обов'язків. Якщо у користувача є критична роль, наприклад адміністратора, чи роль, яка надає доступ до дій що можуть призвести до значного рівня ризику, відповідні ролі повинні переглядатися частіше власниками відповідних ролей та ресурсів.

Системи управління безпечним доступом призначені для автоматизації, візуалізації і спрощення процесу призначення та управління безліччю складних параметрів доступу, описаних вище. Більшість систем управління доступом та інформаційних систем підтримують протокол LDAP, тому доступ до таких активів можна комерційні продукти. Але невеликі підприємства використовують спеціалізоване, застаріле програмне забезпечення або програмне забезпечення що не підтримує протокол LDAP. Тому в такому випадку краще за все використовувати ПЗ для локального відслідковування доступу. Великі компанії використовують ПЗ внутрішньої розробки. Саме цьому є необхідність для створення простої системи для відслідковування доступу для малих підприємств. Тому під час роботи були викладені основні ідеї для управління доступом до локальних застосунків, що буде викладено у практичній частині.

2.3.13. Операційні інструкції для користувачів

Для того, щоб бути впевненим, що користувачі ІТС, спеціалісти та менеджмент підприємства будуть слідувати всім настановам, процесам, процедурам тощо, потрібно створити набір інструкцій по роботі. Відповідні інструкції повинні відноситись до певної авдиторії та покривати певну частину ІТС.

Інструкції для звичайних співробітників повинні бути викладені простою мовою щоб бути максимально зрозумілими для співробітників, що не мають технічних знать. Відповідні інструкції краще за все підтримувати фотографіями, діаграмами, таблицями та іншими візуальними матеріалами.

Інструкції для менеджменту підприємства також повинні бути викладені простою мовою. Відповідні інструкції необхідні лише у випадку, коли необхідно описати специфічні процеси та під процеси, що не можуть бути покриті стандартними інструкціями. Найкращим способом побудови таких інструкцій є доповнення, що посилаються на стандартні інструкції та додатково пояснюють специфічні процеси.

Інструкції для спеціалістів можна писати більш складною мовою, детально описуючі процеси з точки зору адміністратора необхідного процесу чи ресурсу. Це необхідно для того, щоб спеціалісти мали змогу не тільки користуватись певним процесом, але знали як відповідний процес адмініструвати та підтримувати цей процес у робочому стані.

Для того, щоб бути певним, що співробітники достатньо обізнані щодо інструкцій та процесів, необхідно проводити навчальні сесії. Це може бути онлайн лекції, використання онлайн платформ навчання з подальшим тестуванням. Це допоможе вести журнал співробітників що пройшли тестування та контролювати вивчення матеріалу. Потрібно скласти план навчання для груп співробітників та навчати відповідно плану. Також потрібно створити та вести журнал навичок, досвіду тощо, які були отримані під час навчання.

Також непоганим способом тестування та навчання є тестування готових процесів та процедур разом із співробітниками, що відповідають за відповідний процес чи процедуру. Це дозволяє впевнитись, що впроваджені процедури актуальні, а користувачі обізнані роботі у таких предметних областях.

Створення операційних інструкцій регламентовано п. 12.1 стандарту ISO 27001, навчання та тестування персоналу, регламентовано п. 7.2.

2.3.14. Тестова експлуатація СУІБ

Перед тим як починати повноцінну роботу СУІБ потрібно провести достатньо довгий тестовий період. Під час цього періоду будуть проводитися тестування, збір статистики та аналіз потенційних ідей щодо покращення роботи. Під час цього етапу будуть знайдені безліч аспектів, що потребують покращення. Під час цього періоду потрібно створити документи, що будуть свідчити, чи були досягнуті цілі, встановлені під час початку впровадження СУІБ.

Потрібно занотувати, які області ризику були закриті, які типи ризику були зменшені, які корегуючи дії цьому допомогли. Потрібно створити формальний документ, що показує ефективність та результат коригуючих дій та які коригуючі дії були впроваджені. Відповідний документ регламентований п. 10.1 стандарту ISO 27001.

Після першого тестового періоду йду процес доопрацювання СУІБ. Та один чи декілька нових тестовий періодів для того щоб бути впевненим, що створена СУІБ відповідає всім вимогам.

Після закінчення тестового періоду буде зібрана достатня кількість інформації, яка буде надана керівництву для подальшого аудиту та оцінювання. Це необхідно для того, щоб керівництво підприємства підтвердило, що впроваджена СУІБ повністю відповідає бізнес вимогам підприємства та може починати роботу у повному об’ємі. Створення документу, що включає в собі всю інформацію про оцінку ризику, результати моніторингу, зібрану статистику тощо регламентується п. 9.1 стандарту ISO 27001. А результат перегляду менеджментом цієї документації потрібно занести у відгук, що регламентований п. 9.3.

2.3.15. Оперування СУІБ

Після закінчення тестового періоду, потрібно починати оперування СУІБ. Цей процес також включає циклічне тестування та вдосконалення СУІБ. Для цього потрібно створити необхідні процедури, які регламентують процес

підтримки та оновлення СУІБ та процедури циклічного тестування СУІБ. Це дозволить бути впевненим, що СУІБ знаходиться в актуальному стані та працює достатньо добре. Відповідні плани регламентовані п. 17.1 стандарту ISO 27001.

2.4. Фаза сертифікації

Це остаточна фаза процесу створення СУІБ. Відповідна фаза поділяється на 2 етапи: етап попередньої перевірки та оцінювання СУІБ, етап кінцевої перевірки СУІБ. [38] Під час першого етапу перевіряється підготовка системи менеджменту інформаційної безпеки і її документації до сертифікації, оформлюється заявка на проведення сертифікації СУІБ, попередня перевірка і оцінка системи менеджменту інформаційної безпеки, укладається договор на проведення сертифікації системи менеджменту інформаційної безпеки. Під час другої фази готується СУІБ до остаточної перевірки, розробляється програма проведення остаточної перевірки системи, виконувану органом сертифікації, проведення попередньої наради з приводу організації на підприємстві перевірки сертифікації системи, проведення перевірки органом по сертифікації, підготовка попередніх висновків за результатами перевірки для заключної наради, проведення заключного наради, складання звіту про проведення на підприємстві перевірки системи, оформлення, реєстрацію та видачу сертифіката ISO 27001 органом по сертифікації (при позитивному рішенні).

Сертифікат відповідності видається на 3 роки з щорічним аудитом для перевірки роботи СУІБ. Впровадження СУІБ на основі сімейства стандартів ISO 27001 може займати від двох місяців, в залежності від розміру підприємства та характеру бізнес процесів. Вартість сертифікації не менша ніж 15 тисяч гривень. [39]

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1. Короткий опис проблематики

Для того, щоб допомогти підприємству відповідати вимогам стандарту ISO 27001 у сфері контролю доступу, що регламентується п. 9.2 «Управління доступом користувачів: мета полягає у забезпеченні дозволеного доступу користувачів та уникненні несанкціонованого доступу до систем та обладнання», була проаналізована ситуація на підприємстві та визначені основні проблеми.

Загалом управління доступом є частиною загального підходу управління ідентифікацією і доступом (IAM), що полягає у визначенні та управлінні ролями та привілеями доступу окремих сутностей-користувачів до відповідних інформаційних активів. Серед користувачів – клієнти, партнери, сервісні облікові записи (сервісні ОЗ), працівники. Інформаційні активи включають: комп'ютери, смартфони, маршрутизатори, сервери, контролери, датчики тощо. Основною метою IAM є єдина цифрова ідентифікація особи, яка повинна встановлюватися, підтримуватися та контролюватися під час усього життєвого циклу відповідної особи, пристрою тощо. [40]

Сучасні IAM системи представляють з себе сімейство програмних модулів, об’єднаних у єдиний комплекс. Сучасні IAM системи мають модуль для створення та оновлення облікових записів працівників при процесах кадрових змін, управління доступом до електронних ресурсів, управління сервісними ОЗ. Найпопулярніші сучасні IAM системи це рішення від таких провідних компаній CyberArk, Oracle, Okta, Microsoft. [41] Здебільшого відповідні системи продаються за моделлю програмного забезпечення як сервіс (SaaS). Найяскравіший приклад це Microsoft Azure Active Directory, Azure Active Directory External Identities, Azure Active Directory Domain Services. Проте у відповідних рішень є свої обмеження, які обумовлені специфікою SaaS моделлю.

З іншого боку є створені на підприємстві системи внутрішньої розробки. У кожної організації відповідні системи відрізняються через різний контекст бізнес

процесів підприємств. Відповідні системи більш інтегровані у єдину архітектуру підприємства та можуть мати різні програмні модулі.

Щодо підприємств то найкращим способом буде використання SaaS рішень, які можуть за невелику вартість покрити загальну частину задач та системи внутрішньої розробки, які можуть покрити специфічні внутрішні задачі. Відповідними задачами можуть бути: перехід від розрізненого зберігання та відслідковування доступу, відслідковування доступу до сервісних ОЗ, створення різних політик, журналювання наданого доступу до систем із локальною автентифікацією, яких на підприємстві може бути безліч. Аналогічних систем, що можуть виконувати описані дії немає у загальному доступі, кожна організація займається створенням власних систем відслідковування.

Якщо розглядати проблемі підприємств, то можна визначити наступні проблеми. Доступ до більшості систем записується у різні паперові або електроні журнали. Часто відповідні журнали не поєднані один між одним та не підтримуються у актуальному стані. Наступною проблемою є відслідковування доступу сервісних ОЗ до внутрішніх ресурсів, які не можуть контролюватися SaaS системами. Відповідно потрібно відслідковувати доступ сервісних ОЗ через те, що використання таких ОЗ накладає додаткові ризики, пов’язаних із тим, що доступ до відповідних ОЗ можуть мати багато користувачів та пароль часто може бути загальновідомим для великої кількості співробітників. Також наступною проблемою може бути те, що не завжди можна визначити причину наданого доступу. Доступ може видаватися адміністраторами систем поза журналом та це протирічить вимогам стандарту ISO 27001 у сфері журналювання користувацьких дій.

3.2. Реалізація ПЗ

Саме для того, щоб вирішити поставлені питання була проведена робота із створення прикладу сервісу для контролю доступу. Цей сервіс має назву Access

Tracking System та є єдиною довідковою системою, що зібрала в себе ідеї, отримані під час проходження переддипломної практики.

Після запуску програми відкривається вікно вебпереглядача та відкривається головний екран програми, зображений на рисунку 3.1. На головному екрані наявні 3 основні віджети, тобто програмні модулі, та 2 віджети приховані. Основними віджетами програми є віджет для операцій над індивідуальним доступом, віджет перевірки індивідуального доступу та завантаження напередвизначеного доступу. Другорядними віджетами є таблиці, які відображають наданий та заборонений за замовчуванням доступ.

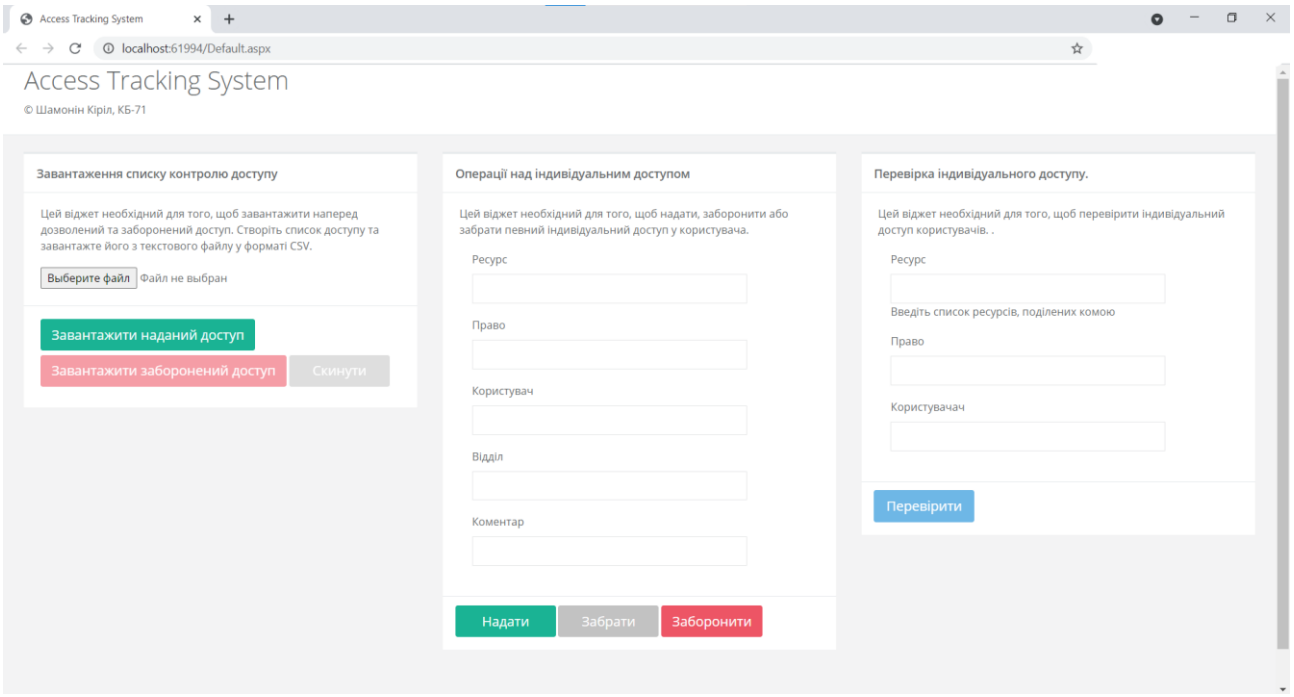
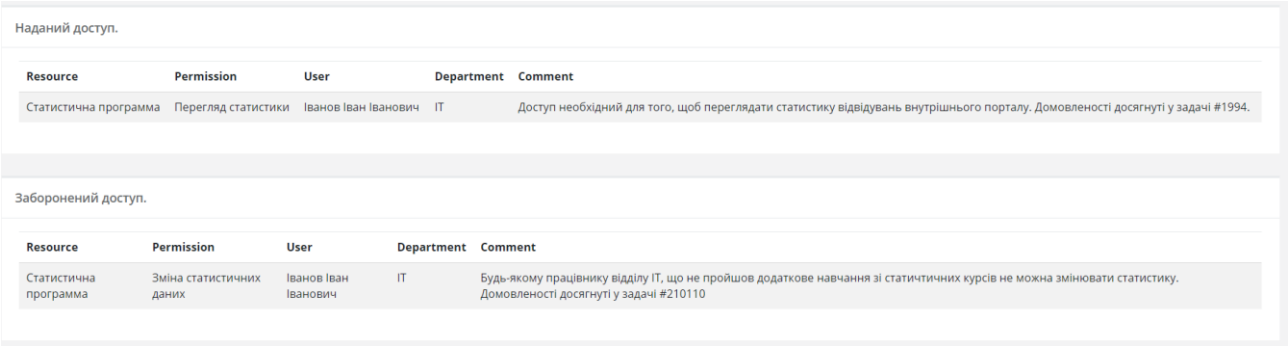


Рисунок 3.1 – Головний екран програми

3.2.1. Віджет операцій над індивідуальним доступом

Відповідний віджет відповідає за формування таблиці контролю доступу. Цей віджет приймає занчення полів «Ресурс» який визначає до якого інформаційного ресурсу потрібно надати, заборонити чи забрати доступ, «Право» визначає яке право потрібно видати для певного ресурсу, «Користувач» це ім’я співробітника що запитує доступ, «Відділ» це відділ користувача що

запитує доступ, «Коментар» це поле в якому потрібно чітко визначити причину надання чи заборони доступу. Можливість надати доступ означає, що особі у списку наданого доступу можна використовувати певний ресурс із певним дозволеним правом. Можливість заборонити доступ реалізує концепцію політик у базовому вигляді. Це означає, що певним групам користувачів або одиничним користувачам наперед заборонено використовувати певний ресурс через певні фактори, які будуть визначені у коментарі. Можливість забрати доступ означає, що у видаляється дані із таблиці, що відносяться до дозволеного або забороненого доступу. Після того, як доступ був дозволений або заборонений, відповідний запис заноситься до відповідної таблиці. Таблиці відслідковування доступу зображені на рисунку 3.2.



Наданий доступ.

Resource	Permission	User	Department	Comment
Статистична программа	Перегляд статистики	Іванов Іван Іванович	IT	Доступ необхідний для того, щоб переглядати статистику відвідувань внутрішнього порталу. Домовленості досягнуті у задачі #1994.

Заборонений доступ.

Resource	Permission	User	Department	Comment
Статистична программа	Зміна статистичних даних	Іванов Іван Іванович	IT	Будь-якому працівнику відділу IT, що не пройшов додаткове навчання зі статичтичних курсів не можна змінювати статистику. Домовленості досягнуті у задачі #210110

Рисунок 3.2 – Таблиці наданого та забороненого доступу.

У роботі використовуються згенеровані онлайн сервісом імена та прізвища, будь-які збіги з іменами та посадами реальних людей є випадковими та не підпадають під вимоги GDPR. [42]

3.2.2. Віджет перевірки індивідуального доступу

Відповідний віджет дозволяє перевірити чи доступ певному користувачу дозволений, заборонений до систем або ще не доданий до системи. Якщо доступ був наданий чи заборонений буде показаний відповідний результат, який зображений на рисунку 3.3 та 3.4, проте якщо доступ не був ані дозволений, ані

заборонений, то за замовчуванням доступ буде заборонений. Повідомлення, що доступ не був ані дозволений, ані заборонений зображене на рисунку 3.5. Це дозволяє реалізувати принципи мінімальної достатності, коли співробітник має можливість отримувати доступ лише до тих інформаційних активів, які потрібні під час роботи, та принцип нульової довіри, який означає те, що все що не дозволено то заборонено. В цьому випадку доступ користувачу заборонений для будь-якого ресурсу, до якого доступ чітко не був дозволений. Для запиту доступу потрібно використовувати систему задач, яка була впроваджена на підприємстві, наприклад Jira.

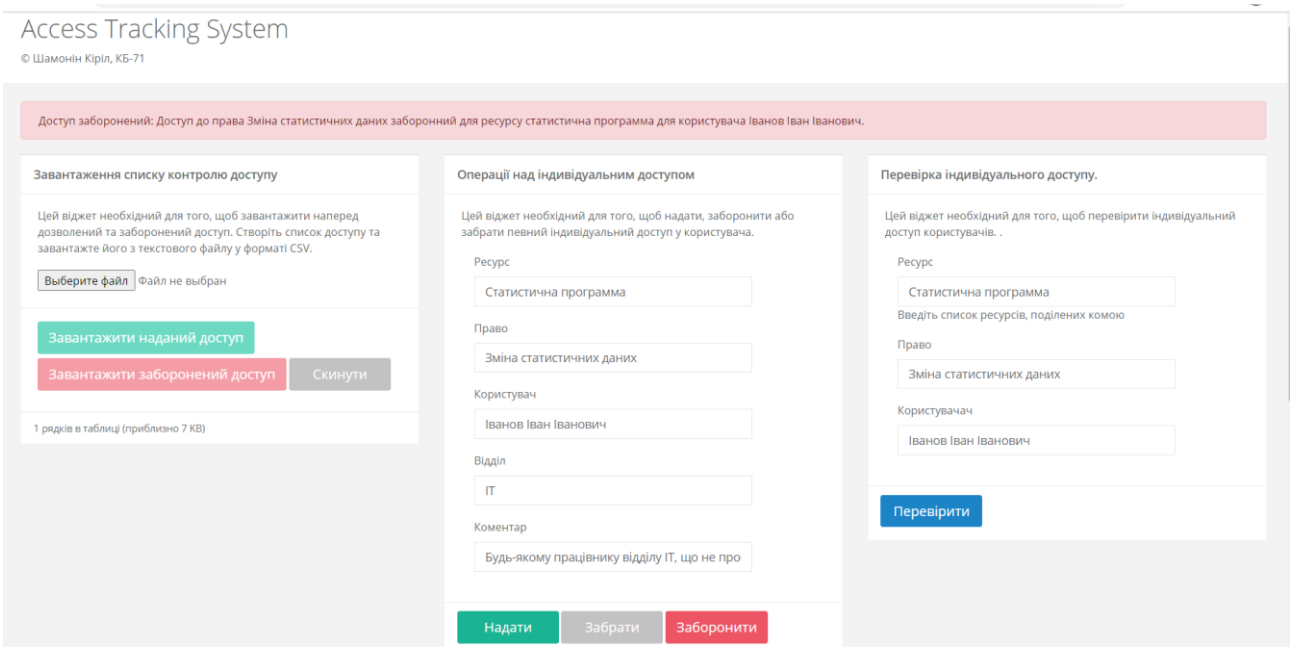


Рисунок 3.3 – Повідомлення, що доступ заборонений

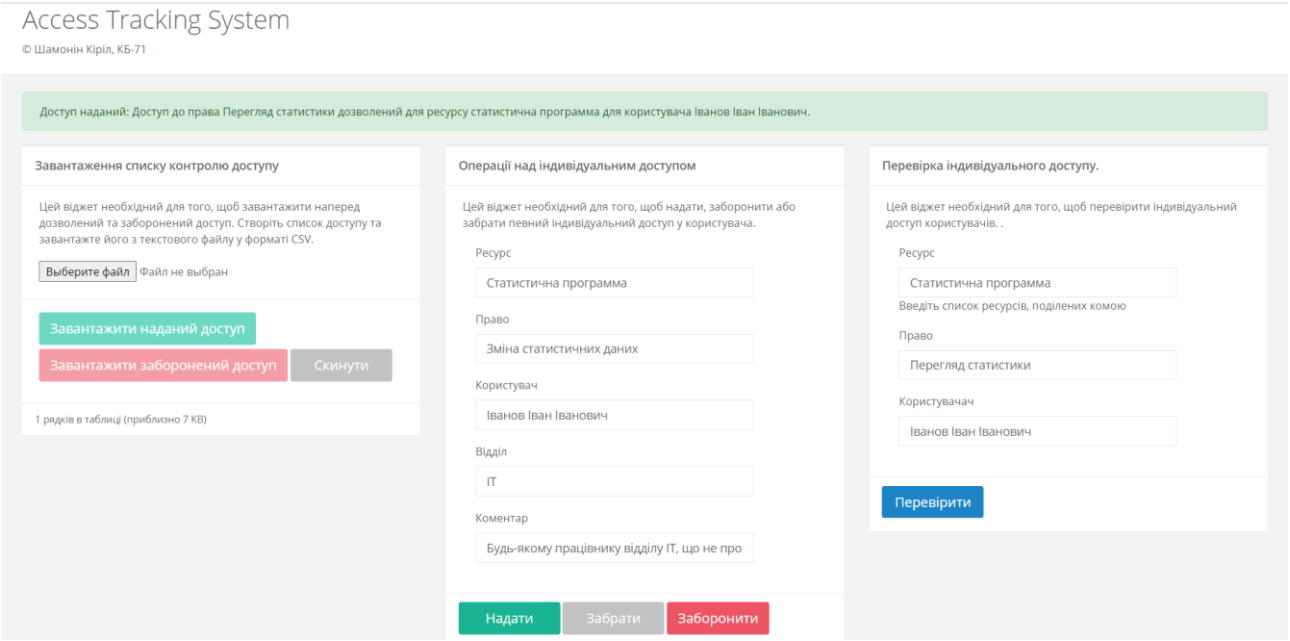


Рисунок 3.4 – Повідомлення, що доступ дозволений

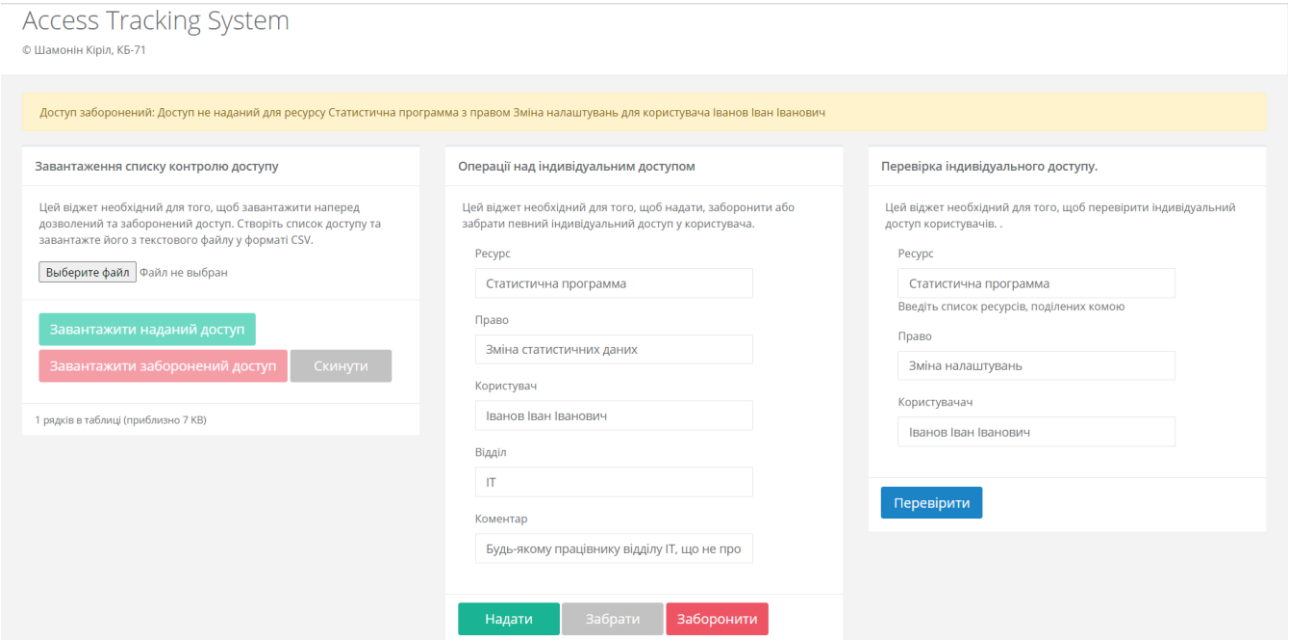
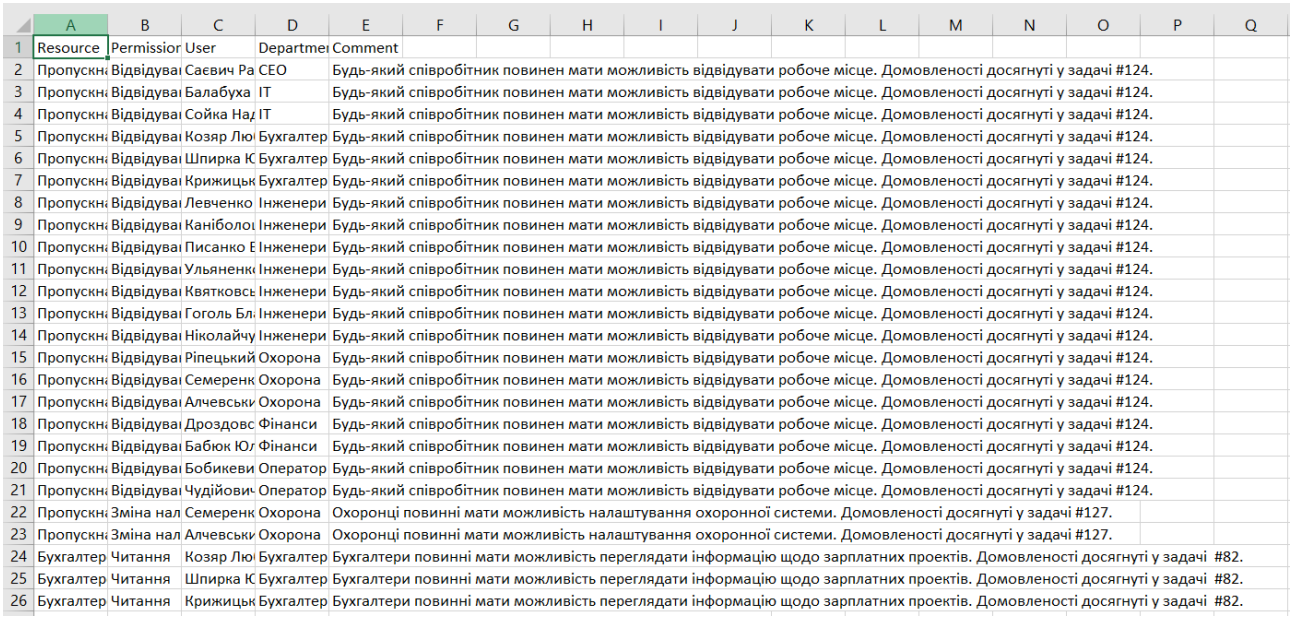


Рисунок 5 – Повідомлення, що доступ не визначений

3.2.3. Віджет завантаження наперед визначеного доступу

Відповідний віджет дозволяє одночасно завантажувати наперед визначений дозволений та/або заборонений доступ, що значно полегшує роботу з програмою. Файли доступу формуються як таблиця у файлі формату CSV, тому

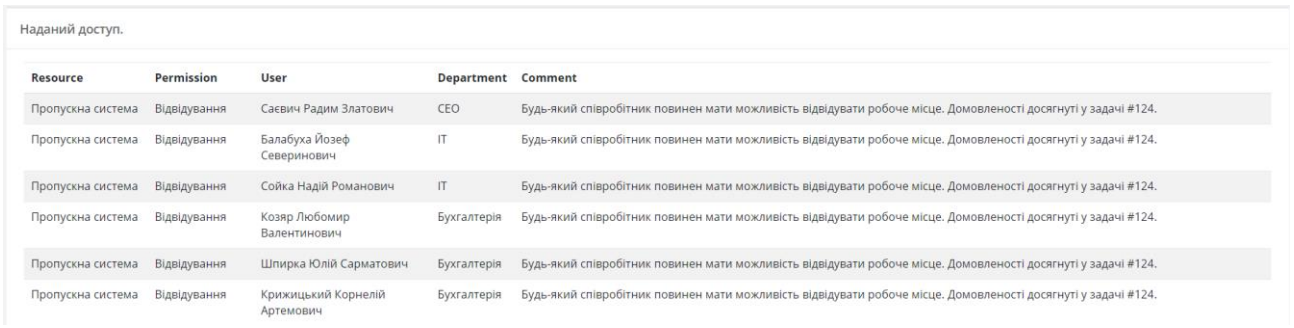
може бути відкритою у будь-якому редакторі. Відповідна таблиця складається з 5 колонок: ресурс, право, користувач, відділ та коментар. Вигляд таблиці зображено на рисунку 3.6.



	A	B	C	D	E
1	Resource	Permissior	User	Departmer	Comment
2	Пропускна	Відвідуваі	Саєвич Ра	CEO	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
3	Пропускна	Відвідуваі	Балабуха	IT	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
4	Пропускна	Відвідуваі	Сойка Над	IT	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
5	Пропускна	Відвідуваі	Козяр Лю	Бухгалтер	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
6	Пропускна	Відвідуваі	Шпирка К	Бухгалтер	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
7	Пропускна	Відвідуваі	Крижицьк	Бухгалтер	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
8	Пропускна	Відвідуваі	Левченко	Інженери	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
9	Пропускна	Відвідуваі	Каніболоі	Інженери	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
10	Пропускна	Відвідуваі	Писанко Е	Інженери	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
11	Пропускна	Відвідуваі	Ульяненкі	Інженери	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
12	Пропускна	Відвідуваі	Квятковсь	Інженери	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
13	Пропускна	Відвідуваі	Гоголь Бл	Інженери	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
14	Пропускна	Відвідуваі	Ніколайчу	Інженери	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
15	Пропускна	Відвідуваі	Ріпецький	Охорона	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
16	Пропускна	Відвідуваі	Семеренк	Охорона	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
17	Пропускна	Відвідуваі	Алчевськи	Охорона	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
18	Пропускна	Відвідуваі	Дроздовс	Фінанси	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
19	Пропускна	Відвідуваі	Бабюк Юл	Фінанси	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
20	Пропускна	Відвідуваі	Бобикеви	Оператор	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
21	Пропускна	Відвідуваі	Чудійови	Оператор	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
22	Пропускна	Зміна нал	Семеренк	Охорона	Охоронці повинні мати можливість налаштування охоронної системи. Домовленості досягнуті у задачі #127.
23	Пропускна	Зміна нал	Алчевськи	Охорона	Охоронці повинні мати можливість налаштування охоронної системи. Домовленості досягнуті у задачі #127.
24	Бухгалтер	Читання	Козяр Лю	Бухгалтер	Бухгалтери повинні мати можливість переглядати інформацію щодо зарплатних проектів. Домовленості досягнуті у задачі #82.
25	Бухгалтер	Читання	Шпирка К	Бухгалтер	Бухгалтери повинні мати можливість переглядати інформацію щодо зарплатних проектів. Домовленості досягнуті у задачі #82.
26	Бухгалтер	Читання	Крижицьк	Бухгалтер	Бухгалтери повинні мати можливість переглядати інформацію щодо зарплатних проектів. Домовленості досягнуті у задачі #82.

Рисунок 3.6 – Приклад файлу, що зберігає наперед визначений наданий доступ

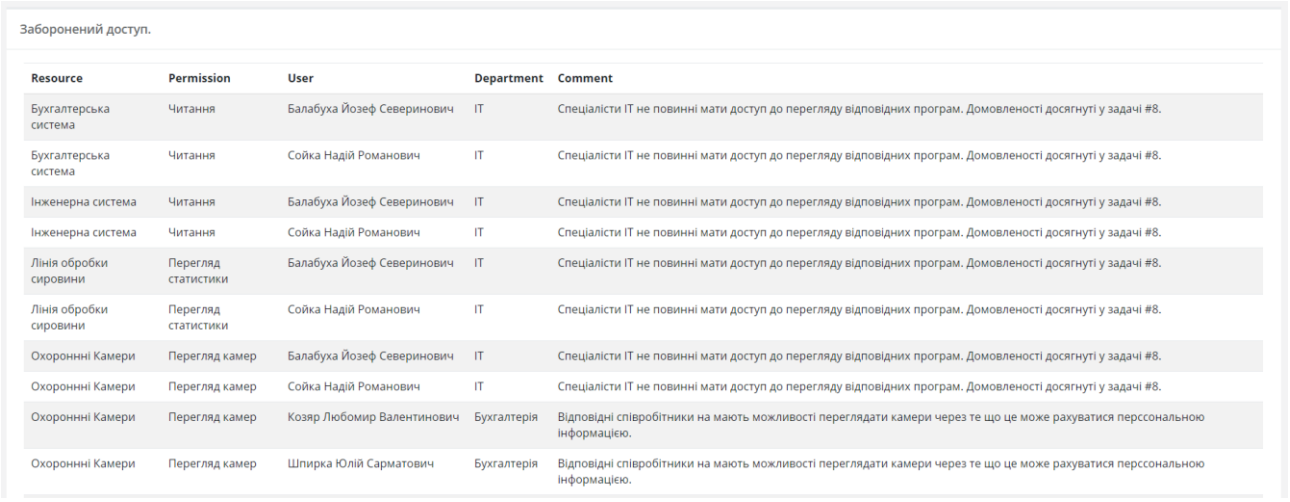
Відповідна таблиця може бути завантажена до програми для економії часових ресурсів оператора. Приклад завантаженої таблиці зображено на рисунку 3.7 та 3.8.



Наданий доступ.

Resource	Permission	User	Department	Comment
Пропускна система	Відвідування	Саєвич Радим Златович	CEO	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
Пропускна система	Відвідування	Балабуха Йозеф Северинович	IT	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
Пропускна система	Відвідування	Сойка Надій Романович	IT	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
Пропускна система	Відвідування	Козяр Любомир Валентинович	Бухгалтерія	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
Пропускна система	Відвідування	Шпирка Юлій Сарматович	Бухгалтерія	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.
Пропускна система	Відвідування	Крижицький Корнелій Артемович	Бухгалтерія	Будь-який співробітник повинен мати можливість відвідувати робоче місце. Домовленості досягнуті у задачі #124.

Рисунок 3.7 – Приклад завантаженої таблиці доступу.



Заборонений доступ.

Resource	Permission	User	Department	Comment
Бухгалтерська система	Читання	Балабуха Йозеф Северинович	IT	Спеціалісти IT не повинні мати доступ до перегляду відповідних програм. Домовленості досягнуті у задачі #8.
Бухгалтерська система	Читання	Сойка Надій Романович	IT	Спеціалісти IT не повинні мати доступ до перегляду відповідних програм. Домовленості досягнуті у задачі #8.
Інженерна система	Читання	Балабуха Йозеф Северинович	IT	Спеціалісти IT не повинні мати доступ до перегляду відповідних програм. Домовленості досягнуті у задачі #8.
Інженерна система	Читання	Сойка Надій Романович	IT	Спеціалісти IT не повинні мати доступ до перегляду відповідних програм. Домовленості досягнуті у задачі #8.
Лінія обробки сировини	Перегляд статистики	Балабуха Йозеф Северинович	IT	Спеціалісти IT не повинні мати доступ до перегляду відповідних програм. Домовленості досягнуті у задачі #8.
Лінія обробки сировини	Перегляд статистики	Сойка Надій Романович	IT	Спеціалісти IT не повинні мати доступ до перегляду відповідних програм. Домовленості досягнуті у задачі #8.
Охоронні Камери	Перегляд камер	Балабуха Йозеф Северинович	IT	Спеціалісти IT не повинні мати доступ до перегляду відповідних програм. Домовленості досягнуті у задачі #8.
Охоронні Камери	Перегляд камер	Сойка Надій Романович	IT	Спеціалісти IT не повинні мати доступ до перегляду відповідних програм. Домовленості досягнуті у задачі #8.
Охоронні Камери	Перегляд камер	Козяр Любомир Валентинович	Бухгалтерія	Відповідні співробітники на мають можливості переглядати камери через те що це може рахуватися перссональною інформацією.
Охоронні Камери	Перегляд камер	Шпирка Юлій Сарматович	Бухгалтерія	Відповідні співробітники на мають можливості переглядати камери через те що це може рахуватися перссональною інформацією.

Рисунок 3.8 – Приклад таблиці забороненого доступу

3.3. Інструкція оператора системи

Для того, щоб правильно користуватись програмою потрібно визначити правила та надати інструкцію для оператора програми. Оператором програми є спеціаліст із контролю доступу, проте якщо на підприємстві недостатня кількість кадрів, то його роль може виконувати спеціаліст з інформаційної безпеки.

План обробки запиту на надання доступу Процес зображений на рисунку 3.9.

1. Формується задача у сервісі для відстежування задач (наприклад, Jira).
2. Оператор програми отримує задачу на надання доступу.
3. Оператор програми визначає чи доступ був наданий раніше або у доступі заборонено. Якщо так – оператор закриває задачу з відповідною приміткою.
4. Якщо доступ не був наданий, тоді оператор отримує підтвердження у менеджера співробітника та власника програми, що доступ може бути наданий до певного користувача. Якщо вони менеджер або власник програми підтвердили, що доступ не може бути наданий, то оператор закриває задачу з відповідною приміткою.

5. Якщо доступ може бути наданий, то оператор надає доступ до відповідної програми, створює обліковий запис для співробітника тощо та заносить дані про наданий доступ до системи Access Tracking System. Після цього закриває задачу з відповідною приміткою.

Рисунок 3.9 – Алгоритм роботи оператора для надання доступу

План обробки запиту на відзив доступу. Процес зображений на рисунку 3.10.

1. Формується задача у сервісі для відстежування задач (наприклад, Jira).
2. Оператор програми отримує задачу на забирання доступу.
3. Оператор задачі перевіряє, чи доступ у відповідного користувача є.
4. Оператор отримує підтвердження у менеджера співробітника, що доступ більше не потрібен. Якщо доступ потрібен – оператор залишає доступ співробітника в системі та закриває задачу з відповідною приміткою.
5. Якщо доступ потрібно забрати, тоді оператор видаляє акаунт працівника у відповідній програмі, видаляє запис із списку наданого доступу у системі Access Tracking System та закриває задачу з відповідною приміткою.

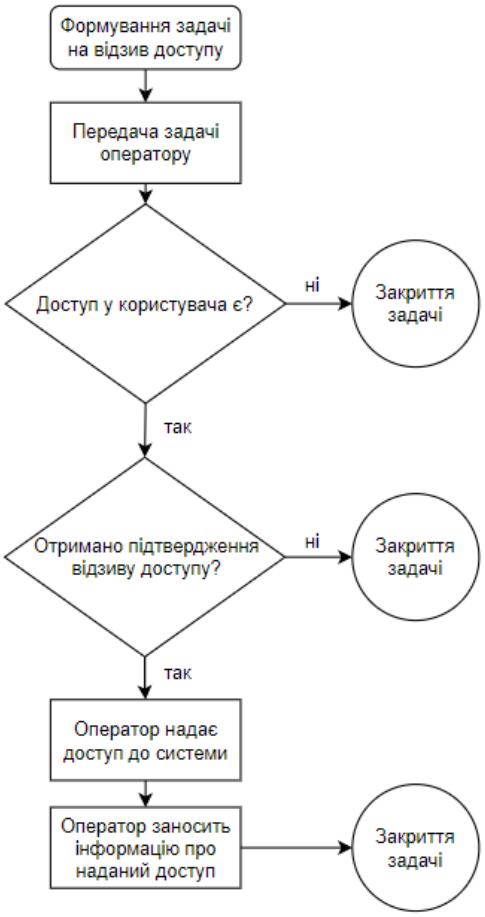


Рисунок 3.10 – Алгоритм роботи оператора для відзиву доступу

ВИСНОВОК

Під час роботи над кваліфікаційною роботою був вивчений важливий матеріал щодо створення системи управління інформаційною безпекою для підприємства. Результати роботи можуть допомогти спеціалістам з інформаційної безпеки під час створення СУІБ підприємства.

Було проаналізоване сімейство стандартів ISO/IEC 27000. Був проаналізований загальний підхід до створення та впровадження СУІБ на підприємстві. Були проаналізовані вимоги до проведення аудитів інформаційної безпеки, проведення аналізу ризиків та способів мінімізації ризику, створення та впровадження процесів та політик для забезпечення, проаналізовані загальні підходи до сертифікації СУІБ за стандартом ISO/IEC 27001.

Розроблене програмне забезпечення дає можливість підприємствам успішно впровадити процес контролю доступу до інформаційних ресурсів із локальною автентифікацією. Це дозволить підвищити загальний рівень безпеки підприємства та відповідати вимогам міжнародних стандартів інформаційної безпеки. Також відповідне програмне забезпечення дозволить менеджменту підприємства зекономити на розробку дорогого програмного забезпечення.

Під час роботи були розглянуті принципи роботи з міжнародними стандартами та сертифікатами інформаційної безпеки, отримані знання програмування мовою C#, використовуючи фреймворк ASP.NET. Отримані практичні знання з написання необхідної документації, впровадження процесів та процедур.

ЛІТЕРАТУРА

1. Манько А. В. СУЧАСНІ ТЕНДЕНЦІЇ ЗАСТОСУВАННЯ ІНТЕРНЕТ-ТЕХНОЛОГІЙ У БІЗНЕСІ : дис. на здобуття наук. ступеня студент : уДК 004.738.5: 334.72 : захист 17.05.2019 / наук. кер. Запащук Л. В. Харків. 5 с.

2. Information Security Fundamentals: книга / за ред. Thomas R. Peltier. Даллас: Auerbach Publications, 2014. 438 с

3. Standardization as a key issue in shared service organization: [Веб-сайт]. URL: https://www.researchgate.net/publication/236270756_Standardization_as_a_key_issue_in_shared_service_organization (дата звернення: 01.06.2021).

4. ISO AND SMALL & MEDIUM ENTERPRISES: [Веб-сайт]. URL: https://www.iso.org/iso-and-smes.html (дата звернення: 17.05.2021).

5. PCI DSS Compliance Guide: UK Costs & Checklist: [Веб-сайт]. URL: https://storekit.com/payments/pci-dss/ (дата звернення: 20.05.2021).

6. Welcome to PCI Compliance Guide: [Веб-сайт]. URL: https://www.pcicomplianceguide.org/faq/ (дата звернення: 19.05.2021).

7. An Introduction to the Components of the Framework: [Веб-сайт]. URL: https://www.nist.gov/cyberframework/online-learning/components-framework (дата звернення: 19.05.2021).

8. A Quick NIST Cybersecurity Framework Summary: [Веб-сайт]. URL: https://cipher.com/blog/a-quick-nist-cybersecurity-framework-summary (дата звернення: 18.05.2021)

9. The PCI DSS (Payment Card Industry Data Security Standard): [Веб-сайт]. URL: https://www.itgovernance.co.uk/pci_dss (дата звернення: 10.05.2019).

10. What is GDPR, the EU’s new data protection law? [Веб-сайт]. URL: https://gdpr.eu/what-is-gdpr (дата звернення: 10.05.2021).

11. What is GDPR? The summary guide to GDPR compliance in the UK: [Веб-сайт]. URL: https://www.wired.co.uk/article/what-is-gdpr-uk-eu-legislation-compliance-summary-fines-2018 (дата звернення: 15.05.2021).

12. What is GDPR? Everything you need to know about the new general data protection regulations: [Веб-сайт]. URL: https://www.zdnet.com/article/gdpr-an-executive-guide-to-what-you-need-to-know/ (дата звернення: 01.05.2021).

13. ISO/IEC 27000:2018. Information technology — Security techniques — Information security management systems — Overview and vocabulary. -, 2018. 27 с.

14. Про прийняття та скасування національних стандартів, прийняття поправок до національних стандартів: [Веб-сайт]. URL: https://zakon.rada.gov.ua/rada/show/v0312774-19?lang=en#Text (дата звернення: 05.05.2021).

15. ISO27001/ISO27002 A Pocket Guide, Second Edition: книга / за ред. Alan Calder. -: IT Governance Publishing, 2013. 86 с.

16. ISO 27001 controls – A guide to implementing and auditing: книга / за ред. Bridget Kenyon. -, 2019. 175 с.

17. ISO 27001 checklist: a step-by-step guide to implementation: [Веб-сайт]. URL: https://www.itgovernance.co.uk/blog/iso-27001-checklist-a-step-by-step-guide-to-implementation (дата звернення: 17.05.2021).

18. SME GUIDE FOR THE IMPLEMENTATION OF ISO/IEC 27001 ON INFORMATION SECURITY MANAGEMENT: [Веб-сайт]. URL: https://www.digitalsme.eu/digital/uploads/SME-Guide-for-the-implementation-of-ISOIEC-27001-on-information-security-management-min-1-1-1.pdf (дата звернення: 05.05.2021).

19. List of mandatory documents required by ISO 27001 (2013 revision): [Веб-сайт]. URL: https://advisera.com/27001academy/knowledgebase/list-of-mandatory-documents-required-by-iso-27001-2013-revision/ (дата звернення: 20.05.2021).

20. A Guide to the Project Management Body of Knowledge, Fourth Edition: книга. -: Project Management Institute, 2011. 459 с.

21. ISO 27001 controls – A guide to implementing and auditing: книга / за ред. Bridget Kenyon. -, 2019. 175 с.

22. Roy, Y. V., Mazur, N. P., & Skladannyi, P. M. (2018). АУДИТ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ – ОСНОВА ЕФЕКТИВНОГО ЗАХИСТУ ПІДПРИЄМСТВА. Електронне фахове наукове видання "Кібербезпека: освіта, наука, техніка", 1(1), 86-93.

23. Common Vulnerability Scoring System SIG: [Веб-сайт]. URL: https://www.first.org/cvss/ (дата звернення: 12.05.2021).

24. ISO 27001 Information Security Management Standard - Clause 7.5: [Веб-сайт]. URL: https://www.mangolive.com/blog-mango/part-17-clause-7.5-documented-information (дата звернення: 25.05.2021).

25. Data Destruction – Are you adhering to industry standards?: [Веб-сайт]. URL: https://diskshred.eu/data-destruction-are-you-adhering-to-industry-standards (дата звернення: 12.05.2021).

26. ISO 27001 – Annex A.11: Physical & Environmental Security: [Веб-сайт]. URL: https://www.isms.online/iso-27001/annex-a-11-physical-and-environmental-security/ (дата звернення: 19.05.2021).

27. Information Security Management: [Веб-сайт]. URL: https://lo.unisa.edu.au/mod/book/view.php?id=743436&chapterid=117761 (дата звернення: 18.05.2021).

28. SUPPLIER SECURITY: [Веб-сайт]. URL: https://www.cpaglobal.com/supplier-security (дата звернення: 20.05.2021).

29. Information Security Incident Management | IS Incident Management: [Веб-сайт]. URL: https://searchinform.com/infosec-blog/2019/04/14/dlp-systems-what-is-a-dlp-system-and-how-does-it-work/Information-Security-Incident-Management/ (дата звернення: 22.05.2021).

30. ISO 27001 Controls: What Is Annex A:16?: [Веб-сайт]. URL: https://bestpractice.biz/iso-27001-controls-what-is-annex-a16/ (дата звернення: 27.05.2021)

31. ISO 27001 Annex: A.17 Information Security Aspects of Business Continuity Management: [Веб-сайт]. URL: https://info-savvy.com/iso-27001-annex-a-17-information-security-aspects-of-business-continuity-management/ (дата звернення: 27.05.2021).

32. 12.3 Information Backup: [Веб-сайт]. URL: https://activaconsulting.co.uk/iso-27002-controls/12-3-information-backup/ (дата звернення: 09.05.2021).

33. What are secure engineering principles in ISO 27001:2013 control A.14.2.5? [Веб-сайт]. URL: https://advisera.com/27001academy/blog/2015/08/31/what-are-secure-engineering-principles-in-iso-270012013-control-a-14-2-5/ (дата звернення: 29.05.2021).

34. User Activity Monitoring and Access Logging Tool: [Веб-сайт]. URL: https://www.solarwinds.com/security-event-manager/use-cases/user-activity-monitoring (дата звернення: 01.04.2021).

35. Information Transfer Policies and Procedures in ISO 27001: [Веб-сайт]. URL: https://iso27001guide.com/information-transfer-policies-and-procedures-in-iso-27001-iso27001-guide-iso27001-guide.html (дата звернення: 12.05.2021).

36. 9.1.1 Access Control Policy: [Веб-сайт]. URL: https://activaconsulting.co.uk/iso-27002-controls/9-1-1-access-control-policy/ (дата звернення: 08.05.2021)

37. Access Management System [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://www.solarwinds.com/access-rights-manager/access-management-system.

38. ISO 27001 - система менеджмента информационной безопасности: [Веб-сайт]. URL: https://ims-cert.com/mezhdunarodnaya-sertifikacziya/iso/iec-27001.html (дата звернення: 27.05.2021).

39. Сертифікат ISO 27001: [Веб-сайт]. URL: https://www.direktiva.com.ua/uk/iso-27001 (дата звернення: 12.05.2021).

40. What is IAM? Identity and access management explained: [Веб-сайт]. URL: https://www.csoonline.com/article/2120384/what-is-iam-identity-and-access-management-explained.html (дата звернення: 15.04.2021).

41. Best IAM Software & Solutions: [Веб-сайт]. URL: https://www.esecurityplanet.com/products/best-iam-software/ (дата звернення: 18.04.2021).

42. Генератор імен та прізвищ онлайн: [Веб-сайт]. URL: https://generator-online.com/uk/names/ (дата звернення: 25.04.2021)

ДОДАТОК А

AccessTrackList.cs

```
namespace AccessTrackingSystem
{
    Ссылок: 3
    public class AccessTrackList : IAccessTrackList
{
    private readonly BaseControlList _granted;
    private readonly BaseControlList _denied;
    Ссылок: 2
    public AccessTrackList()
    {
        _granted = new BaseControlList();
        _denied = new BaseControlList();
    }
    Ссылок: 3
    public string Explain(string[] roles, string permission, string user)
    {   if (_denied.IsIncluded(roles, permission, user))
        {
            var included = _denied.FindIncludedRoles(roles, permission, user);
            return string.Format("Доступ до права {1} заборонний для ресурсу {0} для користувача {2}.", included, permission, user);
        }
        if (_granted.IsIncluded(roles, permission, user))
        {
            var included = _granted.FindIncludedRoles(roles, permission, user);
            return string.Format("Доступ до права {1} дозволений для ресурсу {0} для користувача {2}.", included, permission, user);
        } ;
        return "Дозвіл не виданий для будь-якого користувача!";
    }
        Ссылок: 3
        public bool IsGranted(string[] resource, string permission, string user)
    {
        bool result = false;
            if (!_denied.IsIncluded(resource, permission, user))
        {
            if (_granted.IsIncluded(resource, permission, user))
                result = true;
        }
        return result;
    }
    Ссылок: 2
    public void Grant(string resource, string permission, string user)
    {
        _granted.Include(resource, permission, user);
    }
    ссылка: 1
    public void Revoke(string resource, string permission, string user)
    {
        _granted.Exclude(resource, permission, user);
    }
    Ссылок: 2
    public bool IsDenied(string[] resource, string permission, string user)
    {
        return _denied.IsIncluded(resource, permission, user);
    }
    Ссылок: 2
    public void Deny(string resource, string permission, string user)
    {
        _denied.Include(resource, permission, user);
    }}}
```

AccessTrackListAdapter.cs

```
namespace AccessTrackingSystem
{
    ссылка: 1
    public static class AccessTrackListAdapter
    {
        Ссылок: 2
        private static StringCollection ValidateTable(DataTable table)
        {
            var problems = new CommaDelimitedStringCollection();

            if (!table.Columns.Contains("Resource"))
                problems.Add("Немає колонки: Resource");

            if (!table.Columns.Contains("Permission"))
                problems.Add("Немає колонки: Permission");

            if (!table.Columns.Contains("User"))
                problems.Add("Немає колонки: User");

            if (!table.Columns.Contains("Department"))
                problems.Add("Немає колонки: Department");

            if (!table.Columns.Contains("Comment"))
                problems.Add("Немає колонки: Comment");

            for (var i = 0; i < table.Rows.Count; i++)
            {
                if (table.Rows[i].IsNull("Resource"))
                    problems.Add(string.Format("Немає значення Resource на рядку {0:n0}", i + 1));

                if (table.Rows[i].IsNull("Permission"))
                    problems.Add(string.Format("Немає значення Permission на рядку {0:n0}", i + 1));

                if (table.Rows[i].IsNull("User"))
                    problems.Add(string.Format("Немає значення User на рядку {0:n0}", i + 1));

                if (table.Rows[i].IsNull("Department"))
                    problems.Add(string.Format("Немає значення Department на рядку {0:n0}", i + 1));

                if (table.Rows[i].IsNull("Comment"))
                    problems.Add(string.Format("Немає значення Comment на рядку {0:n0}", i + 1));
            }

            return problems;
        }
        ссылка: 1
        public static void Load(IAccessTrackList acl, DataTable granted, DataTable denied)
        {
            var problems = ValidateTable(granted);
            if ( problems.Count > 0 )
                throw new Exception("Невірна таблиця наданого доступу: " + problems);

            problems = ValidateTable(denied);
            if (problems.Count > 0)
                throw new Exception("Невірна таблиця забороненого доступу: " + problems);
            for (var i = 0; i < granted.Rows.Count; i++)
            {
                var row = granted.Rows[i];

                var Resource = (string) row["Resource"];
                var Permission = (string)row["Permission"];
                var User = (string)row["User"];

                acl.Grant(Resource, Permission, User);
            }
            for (var i = 0; i < denied.Rows.Count; i++)
            {
                var row = denied.Rows[i];

                var Resource = (string)row["Resource"];
                var Permission = (string)row["Permission"];
                var User = (string)row["User"];

                acl.Deny(Resource, Permission, User);
    }}}}
```

BaseControlItem.cs

```
namespace AccessTrackingSystem
{
    Ссылок: 6
    public class BaseControlItem
    {
        private readonly Dictionary<string, StringCollection> _roles;
        ссылка: 1
        public BaseControlItem() { _roles = new Dictionary<string, StringCollection>(); }
        ссылка: 1
        public void Exclude(string resource, string permission)
        {
            resource = StringHelper.Sanitize(resource);
            if (resource == null)
                throw new ArgumentNullException();

            var value = GetValue(permission);
            if (value.Contains(resource))
                value.Remove(resource);
        }
        ссылка: 1
        public CommaDelimitedStringCollection FindIncludedRoles(string[] resources, string permission)
        {
            var includedRoles = new CommaDelimitedStringCollection();

            var key = StringHelper.Sanitize(permission);

            if (key == null || !_roles.ContainsKey(key))
                return includedRoles;

            var value = _roles[key];
            foreach (var res in resources)
            {
                var p = StringHelper.Sanitize(res);
                if (value.Contains(p))
                    includedRoles.Add(p);
            }

            return includedRoles;
        }
        ссылка: 1
        public bool IsIncluded(string[] resources, string permission)
        {
            var key = StringHelper.Sanitize(permission);

            if (key == null)
                return false;

            var value = !_roles.ContainsKey(key) ? null : _roles[key];
            return value != null && resources.Any(role => value.Contains(StringHelper.Sanitize(role)));
        }
        Ссылок: 2
        private StringCollection GetValue(string permission)
        {
            var key = StringHelper.Sanitize(permission);
            if (key == null)
                throw new ArgumentNullException();

            StringCollection value;
            if (!_roles.ContainsKey(key))
            {
                value = new StringCollection();
                _roles.Add(key, value);
            }
            else
            {
                value = _roles[key];
            }

            return value;
        }
    }
}
```

BaseControlList.cs

```
namespace AccessTrackingSystem
{
    Ссылок: 5
    public class BaseControlList
    {
        private readonly Dictionary<string, BaseControlItem> _operations;
        Ссылок: 2
        public BaseControlList() { _operations = new Dictionary<string, BaseControlItem>(); }
        ссылка: 1
        public void Exclude(string resource, string permission, string user)
        {
            resource = StringHelper.Sanitize(resource);
            permission = StringHelper.Sanitize(permission);
            var value = GetValue(user);
            value.Exclude(resource, permission);
        }
        Ссылок: 2
        public CommaDelimitedStringCollection FindIncludedRoles(string[] resource, string permission, string user)
        {
            var value = GetValue(user);
            return value.FindIncludedRoles(resource, permission);
        }
        Ссылок: 2
        public void Include(string resource, string permission, string user)
        {
            resource = StringHelper.Sanitize(resource);
            permission = StringHelper.Sanitize(permission);
            var value = GetValue(user);
            value.Include(resource, permission);
        }
        Ссылок: 5
        public bool IsIncluded(string[] resources, string permission, string user)
        {
            permission = StringHelper.Sanitize(permission);
            var key = StringHelper.Sanitize(user);

            if (key == null)
                return false;

            var value = !_operations.ContainsKey(key) ? null : _operations[key];
            return value != null && value.IsIncluded(resources, permission);
        }
        Ссылок: 3
        private BaseControlItem GetValue(string resource)
        {
            var key = StringHelper.Sanitize(resource);

            BaseControlItem value;
            if (!_operations.ContainsKey(key))
            {
                value = new BaseControlItem();
                _operations.Add(key, value);
            }
            else
            {
                value = _operations[key];
            }
            return value;
        }
    }
}
```

IAccessTrackList.cs

```
namespace AccessTrackingSystem
{
    // This interface provides requirements for the Access Tracking System base logic
    Ссылок: 5
    public interface IAccessTrackList
    {
        Ссылок: 3
        bool IsGranted(string[] resources, string permission, string user);
        Ссылок: 2
        void Grant(string resource, string permission, string user);
        ссылка: 1
        void Revoke(string resource, string permission, string user);

        Ссылок: 2
        bool IsDenied(string[] resources, string permission, string user);
        Ссылок: 2
        void Deny(string resource, string permission, string user);

        Ссылок: 3
        string Explain(string[] resources, string permission, string user);
    }
}
```

CsvHelper.cs

```
namespace AccessTrackingSystem
{
    ссылка: 1
    public class CsvHelper
    {
        Ссылок: 22
        public DataTable Table { get; private set; }

        ссылка: 1
        public void Read(Stream stream, Encoding encoding)
        {
            Read(stream, encoding, ',', 0, int.MaxValue);
        }

        ссылка: 1
        public void Read(Stream stream, Encoding encoding, char chFieldSeparator, int fromLine, int thruLine)
        {
            var lineCount = 0;
            Table = new DataTable();
            using (TextReader reader = new StreamReader(stream, encoding))
            {
                Table.Columns.Add("Column001");

                string sLine;
                while ((sLine = reader.ReadLine()) != null)
                {
                    if (sLine.Length == 0)
                        continue;

                    lineCount++;

                    if (lineCount > 1 && (lineCount < fromLine || lineCount > thruLine))
                        continue;
                    var row = Table.NewRow();
                    Table.Rows.Add(row);

                    var i = 0;
                    var nMode = 0;
                    var nField = 0;
                    var bContinueParsing = true;
                    while (bContinueParsing)
                    {
                        switch (nMode)
                        {
                            case 0: // Search for next entry.
                                {
                                    if (chFieldSeparator == CsvControlChars.Tab)
                                    {
                                        // Don't skip the tab when it is used as a separator.
                                        while (char.IsWhiteSpace(sLine[i]) && sLine[i] != CsvControlChars.Tab)
                                            i++;
                                    }
                                    else
                                    {
                                        while (char.IsWhiteSpace(sLine[i]))
                                            i++;
                                    }
                                    nMode = 1;
                                    break;
                                }
                            case 1: // Determine if field is quoted or unquoted.
                                {
                                    // first check if field is empty.
                                    var chPunctuation = sLine[i];
                                    if (chPunctuation == chFieldSeparator)
                                    {
                                        i++;
                                        nField++;
                                        if (nField > Table.Columns.Count)
                                            Table.Columns.Add("Column" + nField.ToString("000"));
                                        nMode = 0;
                                    }
                                    else if (chPunctuation == '\"')
                                    {
                                        i++;
                                        // Field is quoted, so start reading until next quote.
                                        nMode = 3;
                                    }
                                    else
                                    {
                                        // Field is unquoted, so start reading until next separator or end-of-line.
                                        nMode = 2;
                                    }
                                    break;
                                }
                            case 2: // Extract unquoted field.
                                {
                                    nField++;
                                    if (nField > Table.Columns.Count)
                                        Table.Columns.Add("Column" + nField.ToString("000"));

                                    var nFieldStart = i;
                                    // Field is unquoted, so start reading until next separator or end-of-line.
                                    while (i < sLine.Length && sLine[i] != chFieldSeparator)
                                        i++;
                                    var nFieldEnd = i;

                                    var sField = sLine.Substring(nFieldStart, nFieldEnd - nFieldStart);
                                    row[nField - 1] = sField;
                                    nMode = 0;
                                    i++;
```

```
                        break;
                    }
                case 3: // Extract quoted field.
                    {
                        nField++;
                        if (nField > Table.Columns.Count)
                            Table.Columns.Add("Column" + nField.ToString("000"));

                        bool bMultiline;
                        var sbField = new StringBuilder();
                        do
                        {
                            var nFieldStart = i;
                            // Field is quoted, so start reading until next quote.  Watch out for an escaped quote (two double quotes).
                            while ((i < sLine.Length && sLine[i] != '\"') || (i + 1 < sLine.Length && sLine[i] == '\"' && sLine[i + 1] == '\"'))
                            {
                                if (i + 1 < sLine.Length && sLine[i] == '\"' && sLine[i + 1] == '\"')
                                    i++;
                                i++;
                            }
                            var nFieldEnd = i;
                            if (sbField.Length > 0)
                                sbField.Append(CsvControlChars.CrLf);
                            sbField.Append(sLine.Substring(nFieldStart, nFieldEnd - nFieldStart));


                            bMultiline = (i == sLine.Length);
                            if (bMultiline)
                            {
                                sLine = reader.ReadLine();
                                i = 0;
                                if (sLine == null)
                                    break;
                            }
                        } while (bMultiline);

                        if (sLine != null)
                        {
                            // Skip all characters until we reach the separator or end-of-line.
                            while (i < sLine.Length && sLine[i] != chFieldSeparator)
                                i++;
                        }

                        var sField = sbField.ToString();
                        sField = sField.Replace("\"\"", "\"");
                        row[nField - 1] = sField;
                        nMode = 0;
                        i++;
                        break;
                    }
                default:
                    bContinueParsing = false;
                    break;
            }
            if (i >= sLine.Length)
                break;
        }
    }
}
AssignColumnNames();
}

ссылка: 1
public void AssignColumnNames()
{
    if (Table.Columns.Count <= 0 || Table.Rows.Count <= 1)
        return;

    var row = Table.Rows[0];
    for (var i = 0; i < Table.Columns.Count; i++)
    {
        var column = Table.Columns[i];
        var name = row[i];
        if (name != DBNull.Value)
            column.ColumnName = (string)name;
    }
    row.Delete();
}

Ссылок: 3
internal static class CsvControlChars
{
    internal const string DoubleQuote = "\"\"";
    internal const char Quote = '"';

    ссылка: 1
    internal static string CrLf
    {
        get { return "\r\n"; }
    }

    Ссылок: 2
    internal static char Tab
    {
        get { return '\t'; }
    }
}

}
```

ObjectHelper.cs

```
namespace AccessTrackingSystem
{
    Ссылок: 0
    public static class ObjectHelper
    {

        // the way to calculate the size of managed object!

        Ссылок: 2
        internal class Size<TT>
        {
            private readonly TT _obj;
            private readonly HashSet<object> references;
            private static readonly int PointerSize =
            Environment.Is64BitOperatingSystem ? sizeof(long) : sizeof(int);
            ссылка: 1
            public Size(TT obj)
            {
                _obj = obj;
                references = new HashSet<object>() { _obj };
            }
            ссылка: 1
            public long GetSizeInBytes()
            {
                return this.GetSizeInBytes(_obj);
            }
        ссылка: 1
        public static long SizeInBytes<T>(this T someObject)
        {
            var temp = new Size<T>(someObject);
            var tempSize = temp.GetSizeInBytes();
            return tempSize;
        }
    }
}
```

StringHelper.cs

```
namespace AccessTrackingSystem
{
    Ссылок: 18
    public static class StringHelper
    {
        Ссылок: 14
        public static string Sanitize(string value) { return string.IsNullOrEmpty(value) ? null : value.Trim().ToLower(); }

        Ссылок: 4
        internal static object Quote(string value) { return value.Replace("'", "''"); }
    }
}
```

UploadModel.cs

```
namespace AccessTrackingSystem
{
    Ссылок: 2
    public class UploadModel
    {
        Ссылок: 9
        public string Error { get; set; }
        Ссылок: 6
        public DataTable Table { get; set; }
    }
}
```

Default.aspx

```
<!DOCTYPE html>

<html>
<head runat="server">

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Access Tracking System</title>

     <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet">

    <link href="/Styles/style.css" rel="stylesheet">

</head>
<body>

    <form runat="server">

    <div id="wrapper">

        <div id="page-wrapper" class="gray-bg">

            <div class="row wrapper border-bottom white-bg page-heading">
                <div class="col-lg-12">
                    <h1>Access Tracking System</h1>
                    <ol class="breadcrumb">
                        <li>
                            © Шамонін Кіріл, КБ-71
                        </li>
                    </ol>
                </div>
            </div>

            <div class="wrapper wrapper-content animated fadeInRight">


                <div class="form-horizontal"></div>

                <div runat="server" ID="StatusMessage" Visible="false" class="alert alert-danger" EnableViewState="False"></div>

                <div class="row">

                    <div class="col-lg-4">

                        <div class="ibox float-e-margins">
                            <div class="ibox-title">
                                <h5>Завантаження списку контролю доступу</h5>
                            </div>
                            <div class="ibox-content">
                                <p>
                                    Цей віджет необхідний для того, щоб завантажити наперед дозволений та заборонений доступ. Створіть список  доступу та завантаж
                                </p>
                                <asp:FileUpload runat="server" ID="UploadCsv" />

                            </div>
                            <div class="ibox-content">
                                <asp:Button runat="server" ID="LoadGrantedFromCsv" Text="Завантажити наданий доступ" CssClass="btn btn-w-m btn-primary" />
                                <asp:Button runat="server" ID="LoadDeniedFromCsv" Text="Завантажити заборонений доступ" CssClass="btn btn-w-m btn-danger" />
                                <asp:Button runat="server" ID="ResetGrantedAndDenied" Text="Скинути" CssClass="btn btn-w-m btn-default" />
                            </div>
                            <div class="ibox-footer" runat="server" ID="LoadFooter">
                                <span class="pull-right">

                                </span>
                                <asp:Literal runat="server" ID="TableRowCount" /> рядків в таблиці (приблизно <asp:Literal runat="server" ID="TableSize" /> KB)
                            </div>
                        </div>

                    </div>

                    <div class="col-lg-4">

                        <div class="ibox float-e-margins">
                            <div class="ibox-title">
                                <h5> Операції над індивідуальним доступом </h5>
                            </div>
                            <div class="ibox-content">
                                <p>
                                    Цей віджет необхідний для того, щоб надати, заборонити або забрати певний індивідуальний доступ у користувача.
                                </p>
                                <div class="form-horizontal">
                                    <div class="form-group">
                                        <label class="col-lg-2 control-label">Ресурс</label>
                                        <div class="col-lg-10">
                                            <asp:TextBox runat="server" ID="AtlResource" CssClass="form-control" />
                                        </div>
                                    </div>
                                    <div class="form-group">
                                        <label class="col-lg-2 control-label">Право</label>
                                        <div class="col-lg-10">
                                            <asp:TextBox runat="server" ID="AtlPermission" CssClass="form-control" />
                                        </div>
                                    </div>
                                    <div class="form-group">
                                        <label class="col-lg-2 control-label">Користувач</label>
                                        <div class="col-lg-10">
                                            <asp:TextBox runat="server" ID="AtlUser" CssClass="form-control" />
                                        </div>
                                    </div>
```

```
                                <div class="form-group">
                                    <label class="col-lg-2 control-label">Відділ</label>
                                    <div class="col-lg-10">
                                        <asp:TextBox runat="server" ID="AtlDepartment" CssClass="form-control" />
                                    </div>
                                </div>
                                <div class="form-group">
                                    <label class="col-lg-2 control-label">Коментар</label>
                                    <div class="col-lg-10">
                                        <asp:TextBox runat="server" ID="AtlComment" CssClass="form-control" />
                                    </div>
                                </div>

                            </div>
                        </div>
                        <div class="ibox-footer">
                            <asp:Button runat="server" ID="GrantPermission" Text="Надати" CssClass="btn btn-w-m btn-primary" />
                            <asp:Button runat="server" ID="RevokePermission" Text="Забрати" CssClass="btn btn-w-m btn-default" />
                            <asp:Button runat="server" ID="DenyPermission" Text="Заборонити" CssClass="btn btn-w-m btn-danger" />
                        </div>
                    </div>

                </div>

                <div class="col-lg-4">

                    <div class="ibox float-e-margins">
                        <div class="ibox-title">
                            <h5>Перевірка індивідуального доступу.</h5>
                        </div>
                        <div class="ibox-content">
                            <p>
                                Цей віджет необхідний для того, щоб перевірити індивідуальний доступ користувачів. .
                            </p>
                            <div class="form-horizontal">
                                <div class="form-group">
                                    <label class="col-lg-2 control-label">Ресурс</label>
                                    <div class="col-lg-10">
                                        <asp:TextBox runat="server" ID="CheckResources" CssClass="form-control" />
                                        <span class="help-block m-b-none">Введіть список ресурсів, поділених комою</span>
                                    </div>
                                </div>
                                <div class="form-group">
                                    <label class="col-lg-2 control-label">Право</label>
                                    <div class="col-lg-10">
                                        <asp:TextBox runat="server" ID="CheckPermissions" CssClass="form-control" />
                                    </div>
                                </div>
                                <div class="form-group">
                                    <label class="col-lg-2 control-label">Користувачач</label>
                                    <div class="col-lg-10">
                                        <asp:TextBox runat="server" ID="CheckUser" CssClass="form-control" />
                                    </div>
                                </div>

                            </div>
                        </div>
                        <div class="ibox-footer">
                            <asp:Button runat="server" ID="CheckPermission" Text="Перевірити" CssClass="btn btn-w-m btn-success" />
                        </div>
                    </div>

                </div>

                <div class="col-lg-12">

                    <div class="ibox float-e-margins" runat="server" ID="GrantedPanel">
                        <div class="ibox-title">
                            <h5>Наданий доступ.</h5>
                        </div>
                        <div class="ibox-content">
                            <asp:GridView runat="server" ID="GrantedGrid" AutoGenerateColumns="True" CssClass="table table-striped">

                            </asp:GridView>
                        </div>
                    </div>

                    <div class="ibox float-e-margins" runat="server" ID="DeniedPanel">
                        <div class="ibox-title">
                            <h5>Заборонений доступ.</h5>
                        </div>
                        <div class="ibox-content">
                            <asp:GridView runat="server" ID="DeniedGrid" AutoGenerateColumns="True" CssClass="table table-striped">

                            </asp:GridView>
                        </div></div></div></div></div></div></div></form><ul class="nav metismenu" id="side-menu"></ul>
</body>
</html>
```

Default.aspx.cs

```
namespace AccessTrackingSystem
{
    Ссылок: 2
    public partial class Default : Page
    {
        private const int MaximumRowCount = 500;

        Ссылок: 9
        private DataTable Granted
        {
            get
            {
                object value = Session["Granted"] as DataTable;
                if (value == null)
                    Session["Granted"] = CreateTable();
                return (DataTable)Session["Granted"];
            }
            set { Session["Granted"] = value; }
        }

        Ссылок: 8
        private DataTable Denied
        {
            get
            {
                object value = Session["Denied"] as DataTable;
                if (value == null)
                    Session["Denied"] = CreateTable();
                return (DataTable)Session["Denied"];
            }
            set { Session["Denied"] = value; }
        }
        Ссылок: 2
        void AddNewRow(DataTable table)
        {
            var row = table.NewRow();
            row["Resource"] = AtlResource.Text;
            row["Permission"] = AtlPermission.Text;
            row["User"] = AtlUser.Text;
            row["Department"] = AtlDepartment.Text;
            row["Comment"] = AtlComment.Text;
            table.Rows.Add(row);
        }

        Ссылок: 6
        private void BindPermissions()
        {
            var grantCount = Granted.Rows.Count;
            var denyCount = Denied.Rows.Count;

            LoadFooter.Visible = grantCount > 0;
            LoadGrantedFromCsv.Enabled = grantCount == 0;
            LoadDeniedFromCsv.Enabled = grantCount > 0 && denyCount == 0;
            ResetGrantedAndDenied.Enabled = grantCount > 0;
            CheckPermission.Enabled = grantCount > 0;

            GrantedPanel.Visible = grantCount > 0;
            DeniedPanel.Visible = denyCount > 0;

            if (grantCount > 0)
            {
                TableRowCount.Text = string.Format("{0:n0}", grantCount);
                TableSize.Text = string.Format("{0:n0}", Granted.SizeInBytes()/1024);
                BindPermissionsGrid(GrantedGrid, Granted);
            }

            if (denyCount > 0)
                BindPermissionsGrid(DeniedGrid, Denied);
        }

        Ссылок: 2
        private static void BindPermissionsGrid(GridView grid, DataTable table)
        {
            if (table != null)
            {
                grid.BorderWidth = 0;
                grid.BorderStyle = BorderStyle.None;
                grid.GridLines = GridLines.None;

                grid.DataSource = table;
                grid.DataBind();

                if (table.Rows.Count > 0)
                    grid.HeaderRow.TableSection = TableRowSection.TableHeader;
            }
        }

        Ссылок: 2
        private static void BindPermissionsGrid(GridView grid, DataTable table)
        {
            if (table != null)
            {
```

```
            grid.BorderWidth = 0;
            grid.BorderStyle = BorderStyle.None;
            grid.GridLines = GridLines.None;

            grid.DataSource = table;
            grid.DataBind();

            if (table.Rows.Count > 0)
                grid.HeaderRow.TableSection = TableRowSection.TableHeader;
        }
    }

    Ссылок: 2
    static DataTable CreateTable()
    {
        var table = new DataTable();
        table.Columns.Add("Resource");
        table.Columns.Add("Permission");
        table.Columns.Add("User");
        table.Columns.Add("Department");
        table.Columns.Add("Comment");
        return table;
    }
    Ссылок: 2
    private static UploadModel CreateTableFromCsvFile(FileUpload upload)
    {
        var model = new UploadModel();

        if (!upload.HasFile)
        {
            model.Error = "Please select a file to upload.";
            model.Table = null;
        }
        else
        {
            var csv = new CsvHelper();
            csv.Read(upload.PostedFile.InputStream, Encoding.UTF8);

            if (!csv.Table.Columns.Contains("Role"))
                model.Error = "Немає колонки: Role";

            if (!csv.Table.Columns.Contains("Permission"))
                model.Error = "Немає колонки: Permission";

            if (!csv.Table.Columns.Contains("User"))
                model.Error = "Немає колонки: User";

            if (!csv.Table.Columns.Contains("Department"))
                model.Error = "Немає колонки: Department";

            if (!csv.Table.Columns.Contains("Comment"))
                model.Error = "Немає колонки: Comment";

            if (csv.Table.Rows.Count > MaximumRowCount)
                model.Error = string.Format("Ця невелика программа не призначена для великої кількісті даних. Будь ласка" +
                    " використовуйте не більше ніж {0} рядків.", MaximumRowCount);

            model.Table = csv.Table;
        }

        return model;        }
```