СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет ___інформаційних технологій та електроніки_____

Кафедра _____інформаційних технологій та програмування_____

Пояснювальна записка

до магістерської дипломної роботи

_____магістр_____
(освітньо-кваліфікаційний рівень)

на тему: ___ Дослідження систем інформаційної безпеки в банківській установі_____

Виконав: студент 2 курсу, групи ІСТ-22зм_
126 «Інформаційні системи та технології___
(шифр і назва спеціальності)
_____Кобилецька М.М._____
(прізвище та ініціали)
Керівник ___Захожай О. І._____
(прізвище та ініціали)
Рецензент_____
(прізвище та ініціали)

Київ – 2023 року

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет ___інформаційних технологій та електроніки_____
Кафедра _____інформаційних технологій та програмування_____
Освітньо-кваліфікаційний рівень __магістр_____
Спеціальність 126 «Інформаційні системи та технології»_____
(шифр і назва спеціальності)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТП
_____ д.т.н., доц. Захожай О.І.
(підпис)
« ___ » _____2023 р.

ЗАВДАННЯ

на магістерську дипломну роботу студенту
_____Кобилецька Марія Миколаївна_____
(прізвище, ім'я, по батькові)

1.Тема роботи: Дослідження систем інформаційної безпеки в банківській установі,_____
керівник роботи проф., д.т.н. Захожай Олег Ігорович ,
(вчене звання, науковий ступінь, прізвище, ім'я, по батькові)
затверджені наказом університету від «___»____ 2023року № _____
2. Строк подання студентом роботи: 06 грудня 2023 р._____
3. Вихідні дані до роботи: Матеріали науково-дослідної практики, науково-методична література; дані інтернет-мережі ._____
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Вступ
4.2 Аналітичний огляд питання (огляд публічних джерел інформації)
4.3 Основна частина, в якої висвітлити методи, які будуть використовуватися для реалізації проекту.
4.4 Практична частина – огляд технологій, які використовуються під час реалізації проекту.
4.4 Висновки
4.5 Перелік використаних джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
_____
_____

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання Видав	Завдання прийняв

7. Дата видачі завдання __20 жовтня 2023р._____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Одержання завдання на виконання роботи	20.10.2023	
2.	Укладання і погодження з керівником плану і етапів виконання роботи	24.10.2023	
3.	Узагальнення даних літературних джерел	28.10.2023	
4.	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху виконання завдання	01.11.2023	
5.	Аналіз технічних засобів та існуючих систем	07.11.2023	
6.	Реалізація практичної частини завдання	24.11.2023	
7.	Укладання, оформлення та погодження пояснювальної записки з керівником	05.12.2023	
8.	Здача пояснювальної записки на кафедру	06.12.2023	
9.	Підготовка доповіді та презентації	09.12.2023	

Студент _____ Кобилецька М. М._
(підпис) (прізвище та ініціали)

Керівник роботи _____ _Захожай О. І..
(підпис) (прізвище та ініціали)

Анотація

Дипломна робота складається зі вступу, трьох розділів, загальнихт висновків, списку використаних джерел, додатки і має 64 сторінкок основного тексту, 7 рисунків, 3 таблиця, 14 сторінок додатків. Список використаних джерел містить 14 найменування і займає. Загальний обсяг роботи 78 стор.

Актуальність теми роботи. Актуальність цього питання пов'язана зі зростаючими можливостями інформаційних технологій. Розвиток засобів автоматизації, методів і форм обробки інформації та широке використання персональних комп'ютерів зробили інформацію більш вразливою.В свою чергу, протягом останніх років інформація набула особливого значення, а її роль у суспільстві зросла. Стрімкий розвиток технологій, у тому числі дистанційного банківського обслуговування, підвищив обізнаність про потенційні інформаційні загрози та потребу банківських організацій у створенні ефективних систем інформаційної безпеки.

Об’єкт дослідження. Процес захисту інформаційно-банківської системи

Предметом дослідження є методи та програмні засоби захисту інформаційних систем.

Методи дослідження. Розглядаються сучасні кіберзагрози банківського сектору, аналізуються типові схеми атак на локальні мережі, виявляються слабкі місця в захисті локальних мереж, що реалізуються за допомогою базових програмних та апаратних засобів захисту, а також застосовуються додаткові програмні засоби захисту для підвищення рівня безпеки.

Наукова новизна одержаних результатів. Запропоновані способи захисту програмного забезпечення може бути використано на практиці при побудові комплексного захисту локальних мереж банківських установ. Запропоновані в цьому дослідженні методичні рекомендації допоможуть співробітникам інформаційної безпеки банків визначитися з вибором ефективних програмних засобів для підвищення рівня безпеки вже створених локальних мереж.

Практичне значення одержаних результатів. Завдяки застосуванню розробленої системи захисту інформації стає можливим запобігання витоку інформації та застосування системи оцінки ризиків інформаційної безпеки/конфіденційності.

Abstracts

The thesis consists of an introduction, three chapters, general conclusions, a list of references, an appendix and has 64 pages of main text, 7 figures, 3 tables, and appendix pages. The list of references includes 14 titles and occupies 14 pages. The total volume of the work is 78 pages.

Relevance of the topic. The relevance of this issue is associated with the growing capabilities of information technology. The development of automation tools, methods and forms of information processing and the widespread use of personal computers have made information more vulnerable. In turn, in recent years, information has gained special importance and its role in society has increased. The rapid development of technologies, including remote banking, has increased awareness of potential information threats and the need for banking organizations to create effective information security systems.

Object of research. The process of protecting the information and banking system

The subject of the research is methods and software tools for protecting information systems.

Research methods. We consider modern cyber threats to the banking sector, analyze typical attack schemes on local networks, identify weaknesses in the protection of local networks, which are implemented using basic software and hardware protection, and apply additional software protection to increase the level of security.

Scientific novelty of the results. The proposed methods of software protection can be used in practice when building a comprehensive protection of local networks of banking institutions. The methodological recommendations proposed in this study

will help bank information security officers determine the choice of effective software tools to improve the security of already established local networks.

Practical significance of the results. Thanks to the application of the developed information security system, it becomes possible to prevent information leakage and apply an information security/confidentiality risk assessment system.

Зміст

ВСТУП

Становлення України як правової держави зумовлює необхідність модернізації всіх сторін життєдіяльності суспільства, включаючи правові відносини в економічній та фінансовій сферах, які значною мірою залежать від ефективної діяльності банків.

Оскільки банківські системи сучасних держав не є автономними і тісно взаємопов’язані з системами інших країн і міжнародних банківських організацій, проблема підтримки надійності, безпеки та стабільності банківських операцій входить до сфери внутрішнього регулювання.

Банківські системи надзвичайно важливі, і забезпечення їх інформаційної безпеки є необхідною умовою для функціонування. Через значну цінність інформації, що міститься в банківських базах даних, вимоги до її зберігання та обробки постійно високі.

Стрімкий розвиток інформаційних технологій, розширення глобального інформаційного простору, розповсюдження засобів обміну інформацією та широка комп'ютеризація всіх сфер життя роблять важливим питання безпеки інформаційної інфраструктури.

Забезпечення ефективного захисту інформації є дуже важливим для установ банківського сектору, де щоденно обробляються великі обсяги інформації різного рівня конфіденційності.

У більшості випадків ця інформація стає предметом дій конкурентів, що ще більше загострює проблему її захисту від незаконного використання та несанкціонованого доступу.

РОЗДІЛ 1. . ІНФОРМАЦІЙНА БЕЗПЕКА БАНКІВ ТА АНАЛІЗ ЗАГРОЗ ДЛЯ ЇХ ДІЯЛЬНОСТІ

1.1 Поняття інформаційної безпеки банківської установи

На фоні повномасштабного вторгнення в Україну цифрові атаки стали невід'ємною частиною війни як на державні сайти, так і на сайти великих, важливих компаній у системі. Водночас кіберзлочинці освоюють нові методи кібератак, тому обов'язком регуляторів є залишатися систематичним у боротьбі з кібератаками, а банківської галузі – інвестувати в кібербезпеку. На жаль робота банківської системи, посилена умовами війни, в напрямі забезпечення кіберзахисту спрямована на інвестування коштів у пошук засобів захисту даних та рахунків своїх клієнтів та подолання наслідків здійснених кібератак.

За даними НБУ, у 2022 р. майже всі кібератаки, спрямовані на банківський сектор, здійснюються хакерськими групами, які підтримуються урядом країни-окупанта (хакерська група APMageddon, Fancy Bears та інші).

Хронологія	Атаки
01.2022	Whispergate, DDoS, State-sponsored hacking Group
02.2022	Panic Attack, BGP Hijack, Meris, HermeticWiper, DDoS, State-sponsored hacking group
03.2022	Deface, CaddyWiper, DDoS, State-sponsored hacking group
04-05.2022	DDoS - Hacktivist
04-05.2022	Шахрайство – мотивовано фінансами

Наразі кількість кібератак з боку країни-агресора зменшилась за двома напрямками: DDoS-атаки різного характеру, що стосуються всієї банківської системи, крім НБУ, та фішингові атаки різного типу (різні типи шахрайства). Майже усі фішингові атаки, які спрямовані на банківську систему, є виманюванням коштів у клієнтів банків за різними схемами надання допомоги.

Шахраї використовують найпростішу соціальну інженерію, найпростіші методи для створення підроблених мобільних додатків і підроблених банківських сторінок, використовуючи особи справжніх банків.

Інформаційна безпека в банківському секторі стосується безпеки усіх даних, включаючи паперові документи, аудіоінформацію, банківську таємницю, цензуру, фізичну безпеку, безперервність бізнесу, соціальну інженерію тощо.

Найбільшою загрозою для кібербезпеки є людська помилка.

Зрештою, вас можуть обманом змусити розкрити конфіденційну інформацію, не захищені належним чином паролі, використовувати слабкі облікові дані, натискати на зловмисні посилання або відкривати підозрілі вкладення з листів електронної пошти.

Саме люди піддають ризику свої дані та системи, роблячи певні дії (85% порушень, 94% усіх заражених файлів і програм спричинено листами надіслані на електронну пошту)

З огляду на те, що банківська діяльність значною мірою залежить від надійності використовуваних інформаційних технологій, забезпечення інформаційної безпеки стає одним з основоположних принципів функціонування банківської системи в цілому. Одним з основних напрямів забезпечення інформаційної безпеки в банківських установах є захист банківської таємниці.

У структурі інформаційної безпеки банківської установи виділяють такі основні складові:

– безпека інформаційних ресурсів;

– безпека інформаційної інфраструктури;

– безпека «інформаційного поля».

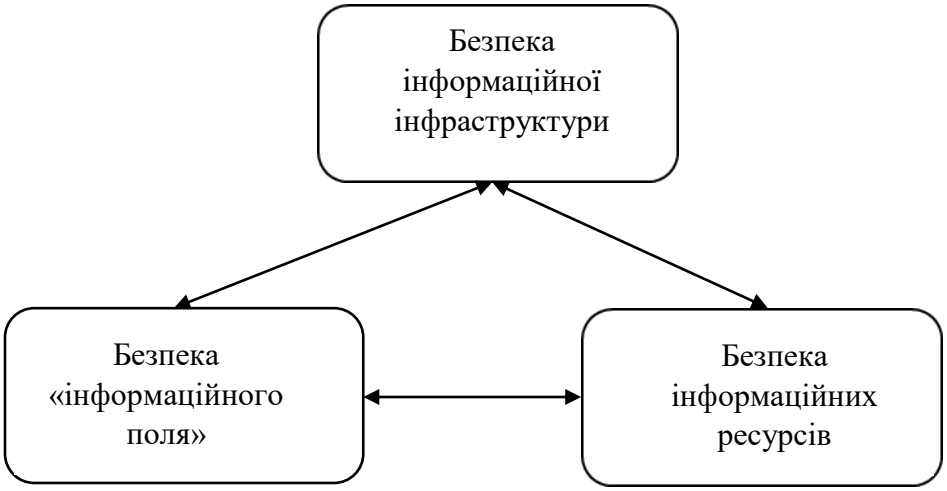


Рис1.1. Складові інформаційної безпеки

Інформаційні ресурси банківської установи - це взаємопов'язані, упорядковані, систематизовані та зафіксовані на матеріальних носіях відомості банківської установи. Тому безпека інформаційних ресурсів - це захищеність такої інформації від несанкціонованого поширення, використання та порушення конфіденційності.

Безпека інформаційної інфраструктури - це стан захищеності електронно-обчислювальних машин (комп'ютерів), систем, комп'ютерних мереж та мереж електрозв'язку банківських установ, за якого гарантується цілісність і доступність інформації, що в них обробляється (зберігається або циркулює).

Безпека "інформаційного сектору" банківської організації ґрунтується на контрольованості здебільшого несистематизованого потоку інформації, що публікується різними учасниками інформаційних відносин, включаючи видавців, друковані ЗМІ, інтернет-видання, конкурентів, органи державної влади та місцевого самоврядування. [2]

Інформаційна безпека в будь-якій організації базується на системі заходів безпеки, реалізованих відповідно до вимог безпеки.

Основними джерелами вимог інформаційної безпеки для організацій є:

- результати оцінки ризиків організації з урахуванням загальної бізнес-стратегії та цілей (під час оцінки ризиків визначаються загрози ресурсам СУІБ, оцінюються вразливості та ймовірність подій, визначається спектр можливих впливів);

- правові вимоги, визначені законами, контрактами та угодами між організацією та її партнерами;
- унікальні принципи обробки інформації, цілі та бізнес-вимоги, розроблені організацією для підтримки своїх функцій.

Методичні рекомендації щодо впровадження системи управління інформаційною безпекою та методології оцінки ризиків відповідно до стандартів Національного банку України визначені як джерело вимог до інформаційної безпеки:

- закони України;
- нормативно-правові акти Національного банку України;
- вимоги платіжних систем та систем переказу коштів;
- внутрішні нормативні документи банку;
- умови угод та договорів з третіми сторонами тощо.

Особливу увагу слід приділяти умовам договорів та контрактів з третіми особами: Відповідно до пункту 6.2 Стандарту НБУ 65.1 СУІБ 2.0:2010, безпека інформації та засобів обробки інформації банку не повинна ставитися під загрозу внаслідок надання продуктів або послуг зовнішніми сторонами. У разі необхідності роботи із зовнішніми сторонами або отримання чи надання продуктів чи послуг від зовнішніх сторін, які можуть потребувати доступу до інформації або засобів обробки інформації банку, банк має здійснити оцінку ризиків для визначення вимог безпеки та наслідків порушення безпеки.

Важливо зазначити, що вимоги до інформаційної безпеки платіжних систем та систем переказу коштів можуть відрізнятися від вимог Банку України, оскільки вони встановлюються клірингловими організаціями платіжних систем та систем переказу коштів (крім систем електронних платежів та національних систем масових електронних платежів, де клірингловою організацією є Банк України).

Вплив ключових сервісів інформаційної безпеки оцінюється за бізнес-процесами/банківськими продуктами, програмно-технологічним комплексом банку. Зауважимо, що однаковий ризик втрати базових послуг безпеки може

бути виявлений у різних бізнес-процесах/банківських продуктах. Це свідчить про наявність певних прогалин у забезпеченні інформаційної безпеки в банках. У цьому випадку необхідно вжити відповідних заходів щодо всіх бізнес-процесів,банківських продуктів для пом'якшення виявлених ризиків інформаційної безпеки.

Нормативно-правові засади забезпечення інформаційної безпеки в Україні становлять:

1. Закон України «Про інформацію»;
2. Закон України «Про доступ до публічної інформації»;
3. Закон України «Про захист інформації в інформаційно-телекомуникаційних системах»;
4. Закон України «Про електронний документ і електронний документообіг»;
5. Закон України «Про електронний цифровий підпис»;
6. Закон України «Про захист персональних даних».

Основи забезпечення інформаційної безпеки в банківській діяльності визначаються:

- Законом України «Про банки і банківську діяльність;
- Законом України «Про Національний банк України»;
- Законом України «Про платіжні системи та переказ коштів в Україні;
- Нормативно-правовими актами Національного банку України.

1.2 Сучасні загрози та ризики для інформаційної безпеки банківської установи

Забезпечення інформаційної безпеки в банківських установах є одним з першочергових завдань, і не потрібно бути експертом, щоб зрозуміти важливість цього питання. Враховуючи, що кожен з нас так чи інакше стикався з банківськими операціями не один-два рази у своєму житті, потрібно особливо ретельно перевіряти рівень захисту такої інформації.

Однак для побудови збалансованої системи інформаційної безпеки необхідно спочатку проаналізувати ризики для систем інформаційної безпеки у банківській сфері.

Проблема починається з неправильного розуміння порушень безпеки інформаційного банку в таких категоріях, як «загроза», «ризик», «джерело загрози», «фактор загрози», «вразливість» і «негативні прояви», «шкідливі фактори впливу» і «перешкоди».

Спільним для них є те, що категорія «небезпека» характеризується протилежно «безпеці».

Міжнародні стандарти управління інформаційною безпекою серії ISO 27000, дотримання яких є обов'язковим у банківській системі України, щодо інформаційної безпеки організації використовують такі основні терміни і поняття:[1]

- інформаційна безпека (information security) – збереження конфіденційності, цілісності та доступності інформації; крім того, можуть враховуватися інші властивості, такі, як автентичність, спостережність, неспростовність та надійність (при цьому для банків України автентичність, спостережність, неспростовність, надійність та автентифікація користувачів та інформаційних ресурсів є обов`язковими вимогами інформаційної безпеки);
- засоби оброблення інформації (information processing facilities) – будь-яка система оброблення інформації, послуга чи інфраструктура, чи місце, де вони фізично розміщені (для банків України засобами оброблення інформації можуть бути власні програмно-технічні комплекси або автоматизовані робочі місця державних/міжнародних платіжних/інформаційних систем);
- система управління інформаційною безпекою (СУІБ) (information security management system ISMS) – частина загальної системи управління, яка ґрунтується на підході, що враховує бізнес-ризики, призначена для розроблення, впровадження, функціонування, моніторингу, перегляду, підтримування та вдосконалення інформаційної безпеки;

•подія інформаційної безпеки (information security event) – ідентифікована подія системи, служби або мережі, яка вказує на можливе порушення політики інформаційної безпеки або відмову засобів захисту чи раніше невідому ситуацію, яка може мати відношення до безпеки;

•інцидент інформаційної безпеки (information security incident) – одна або серія небажаних чи непередбачуваних подій інформаційної безпеки, що мають значну ймовірність компрометації бізнес-операцій і загрози інформаційній безпеці;

•загроза (threat) – потенційна причина небажаного інциденту, який може призвести до шкоди для системи або організації;

•вразливість (vulnerability) – слабкість ресурсу СУІБ або групи ресурсів СУІБ, якою можуть скористатися одна або більше загроз;

•ризик (risk) – комбінація ймовірності події та її наслідку (ризиком інформаційної безпеки банку вважається ймовірність того, що визначена загроза, впливаючи на вразливості ресурсу або групи ресурсів, може спричинити шкоду банку);

•оцінювання ризику (risk evaluation) – процес порівняння кількісно оціненого ризику із заданими критеріями ризику для встановлення його значимості;

•управління ризиком (risk management) – скоординовані дії в організації щодо регулювання та контролю ризику (управління ризиком зазвичай містить оцінку ризику, оброблення ризику, прийняття ризику і доведення ризику до відома);

•заходи безпеки (control) – засоби управління ризиком, які включають політику, процедури, настанови, практику або організаційні заходи, які можуть бути адміністративного, технічного, управлінського або правового характеру;

•політика (policy) – загальні наміри та вказівки, затверджені керівництвом.

Основними завданнями системи інформаційної безпеки є:

– виявлення та усунення загроз безпеки нанесенню економічного, фінансового, матеріального та морального збитку;

– створення механізмів реагування на загрози розвитку і функціонуванню підприємства та національній безпеці;

– прийняття заходів щодо забезпечення безпеки персоналу підприємства та інше.

В інформаційних відносинах банку можуть виникати два типи загроз:

1. загрози, пов'язані з порушенням цілісність інформаційних ресурсів (доступ до яких обмежений) – загрози інформації ;

2. загрози, що виникають внаслідок формування інформаційного середовища (умов) такої організації – інформаційні загрози.

Всі загрози можна згрупувати наступним чином:

1) випадкові загрози: помилки, а також події, що не залежать від людини (природні явища або викликані діяльністю людини);

2) навмисні загрози: можуть реалізуватися учасниками процесу обробки інформації (копіювання і крадіжка програмного забезпечення; несанкціоноване введення даних; зміна або знищення даних на магнітних носіях; крадіжка інформації; несанкціоноване використання ресурсів комп'ютерів; несанкціоноване використання банківських автоматизованих систем; несанкціонований доступ до інформації високого рівня секретності; знищення інформації);

3) перекручення інформації: зміна її змісту, порушення її цілісності, в тому числі і часткове знищення.

Найбільшою загрозою для кібербезпеки є людська помилка. Саме люди зрештою піддають ризику дані та системи через те, що їх обманом змусили надати конфіденційну інформацію, не захистили належним чином свої паролі, використали слабкі облікові дані, натиснули шкідливі посилання або відкрили підозрілі вкладення електронної пошти (85% порушень кібербезпеки є наслідком людської помилки, 94% всіх заражених файлів та програм передаються через електронну пошту).

Безпека банківських інформаційних технологій насамперед пов'язана із захистом від хакерів, вірусів, спаму, фішингу та інших загроз з Інтернету.

Встановлення вимог до комп'ютерів і комунікаційного обладнання та інформації, яку вони зберігають, обробляють і передають, забезпечує цілісність, доступність і конфіденційність банківської інформації.

Хоча інформаційна безпека в банківському секторі значною мірою базується на нормативно-правових актах та вимогах національної безпеки, організація безпеки банківських інформаційних технологій покладається на органи управління та технічне забезпечення банківської системи, а отже, реалізація захисту відбувається через організаційні та технічні елементи банківської діяльності. Безпека інформаційних технологій в банках базується на ефективному управлінні безпекою банківських процесів, включаючи використання кіберстрахування, дотримання нормативних вимог до банківської безпеки, надання гарантій безпеки та забезпечення безперервності банківських операцій шляхом виявлення потенційних кіберзагроз. Все це вимагає компетентності керівництва банку, фінансових працівників, економістів, аналітиків, маркетологів та юристів, які використовують економіко-математичні методи. У банківській IT-безпеці все залежить від процесів управління ризиками.

Банки зобов’язані уважно стежити за потенційними загрозами та ризиками. При цьому необхідно чітко розрізняти обёєкти кібератак. Групуючи найбільш типові кібератаки на банківський сектор, можна виділити такі елементи: конфіденційна або банківська таємниця, банківська інфраструктура, кошти клієнтів і банку, веб-сайти банків і регуляторів.

Дослідження Базельського комітету з банківського нагляду з аналізу впливу фінтеху на банківську діяльність (у тому числі трансформації банківських ризиків) ключовими ризиками, пов’язаними з розвитком цифрових технологій, визначає стратегічний ризик, операційний ризик, кіберризик, комплаєнс-ризик.

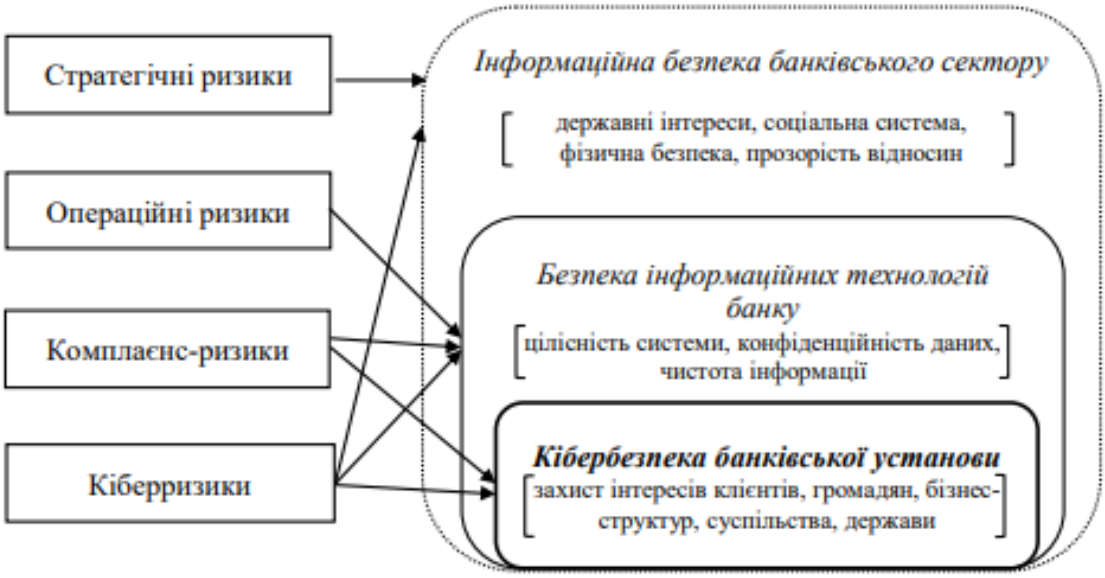


Рис..1.2. Кібербезпека в системі інформаційної безпеки банківського сектору

Фактор ризику зі стратегічних причин є великим і може передбачити потенційні загрози, включаючи вплив фінансових втрат на банківські операції. Ризик втрати прибутковості через тривалий період адаптації зростає, оскільки банки відчувають стратегічні ризики від стрімкого розвитку технологій нових банківських продуктів, доступних у всьому світі. Адаптація до нових продуктів у кіберпросторі банківського сектору для окремих банків триватиме довше, ніж для учасників ринку, які пропонують аналогічні банківські послуги та мають більшу клієнтську базу. Відсутність гнучкості у спілкуванні з клієнтами може призвести до зниження прибутковості банківського сектору, що потенційно може зашкодити здатності існуючих установ виживати в періоди напруженої ділової активності. Зниження значущості суспільної системи призведе до зриву процесів реалізації державних інтересів.

Існують операційні ризики, пов’язані з використанням електронних систем у банківському секторі, які взаємопов’язані в межах окремої банківської установи та охоплюють десятки банків. Це означає, що якщо використана електронна платформа та сервер вийдуть з ладу, вся банківська IT-інфраструктура також вийде з ладу, підриваючи відсутність досвіду управління ризиками, в якому сьогодні працюють окремі банківські установи, що робить

дедалі складнішим (і дорогим) усунення наслідків кіберінциденту . Крім того, реформувати та адаптувати старі банківські IT-системи складніше. Взаємовідносини між банками та спеціалізованими компаніями створюють значні ризики для операційних ризиків банківської системи, що ускладнює запобігання та подолання порушень норм фінансового моніторингу. Це призводить до збереження конфіденційності та чистоти інформації. Забезпечення заходів кібербезпеки банків зумовлює необхідність залучення юридичних спеціалізованих IT-компаній, що є необхідним через обмеженість ресурсів.

Незважаючи на те, що деякі банківські функції можуть бути передані аутсорсингу, ризики та зобов'язання все ще несуть банки під керівництвом Базельського комітету з банківського нагляду.[3]

Поширеність кіберризиків у банківському секторі зростає через зростання поширеності та поширення цифрових атак, сервісів функціональних моделей, зараження даних клієнтів та звітних даних про діяльність банків. Кібербезпека всередині банку є обов'язком керівництва та спеціалістів. Відсутність належної кібербезпеки в банку може призвести до втрати цілісності даних, несанкціонованого доступу до даних клієнтів та можливих збоїв технічної системи, які можуть завдати шкоди довірі клієнтів, суспільства чи держави.

Ризики комплаєнсу можуть призвести до порушень законодавчих норм, включаючи збитки, додаткові збитки та втрату репутації через співпрацю кількох компаній, які прагнуть отримати доступ до персональних даних клієнтів, що порушує правила чесної конкуренції, корпоративну етику, конфіденційність або цілісність інформації.

Кіберризики безпосередньо впливають на інформаційну систему та безпеку держави, роблячи банківську систему вразливою до цих ризиків цифровізації.

1.3 Висовки до розділу

У першій частині ми проаналізували ризики та загрози існуванню інформаційної безпеки банківської установи.

Загалом, чітка концепція "загрози" вимагає подальшого вивчення та розробки, і основна увага повинна бути зосереджена на створенні ефективних та реалістичних систем для моніторингу та управління іншими інформаційними загрозами.

Стратегічна місія банку полягає у забезпеченні інформаційної безпеки для запобігання поточним та потенційним загрозам інформаційній безпеці та наданні механізмів для їх усунення. Інформаційна індустрія, послідовна систематична діяльність передбачає ряд заходів, що забезпечують належне дотримання національних інтересів, державних і правоохоронних органів. Відповідні інтереси людини і суспільства, запобігання нестачі знань і їх швидке вирішення. Враховуючи активну глобалізацію інформаційно-комунікаційних мереж, співпраця важлива не тільки для банків і держав, а й для міжнародних організацій у боротьбі з агресією різних держав.

РОЗДІЛ 2. СТРУКТУРА ТА ЗАХИСТ МЕРЕЖІ БАНКІВСЬКОЇ УСТАНОВИ

2.1 Побудова архітектурної моделі інформаційної банківсьеої моделі

Побудова та підтримка ефективної системи управління банківськими процесами - один із стратегічних пріоритетів розвитку комерційних банків. Вирішення проблеми ефективного управління багато в чому залежить від можливостей інформаційно-банківської системи (ІБС), яка діє в банку. І якість його роботи безпосередньо залежить від архітектурної моделі, на якій побудована система.

Обробка інформації базується на інформаційних технологіях, тобто технологіях, пов’язаних зі збором, обробкою, передачею та використанням інформації. Найчастіше говорять про комп'ютерні інформаційні технології, тобто обробку інформації за допомогою ЕОМ. Інструменти обробки – це зазвичай інструменти прикладного програмного забезпечення, які можна використовувати для виконання певних наборів операцій та дій над інформацією.

У моделі банку виокремлюють три основні рівні управління:

- тактичний;
- оперативний;
- стратегічний.

Обов'язки керівництва різняться на цих рівнях. Основне завдання — проведення заходів на тактичному рівні, які можуть тривати від десяти днів до кількох місяців, крім того,реалізація планів.

Управління на оперативному рівні прагне підвищити ефективність управління на його попередньому тактичному рівні. У той же час шкала часу рівня трохи довша від місяця до цілого року.

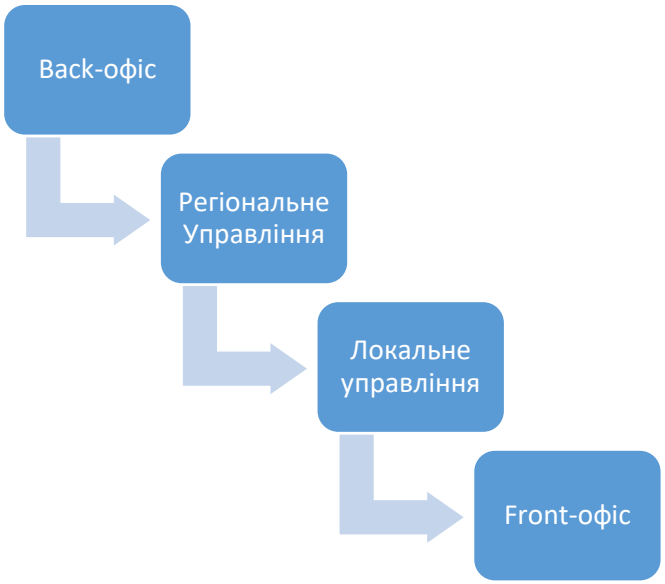
Менеджмент стратегічного рівня має більш самостійні цілі та завдання. Внутрішні та зовнішні цілі поділяються на категорії, і внутрішні цілі зазвичай

стоять перед зовнішніми. Головне завдання стратегічного рівня – координація та узгодження діяльності всіх підрозділів для досягнення зовнішніх цілей. Зовнішні цілі банків завжди були і будуть унікальними.

Основний принцип побудови архітектури інформаційної системи управління банком - "кожному своє". Це означає, що кожен учасник процесу прийняття рішень, їх реалізації та контролю має доступ до необхідної інформації, а також вимогам актуальності та надійності в рамках загального інформаційного простору.

Нова категорія знань-це не те саме, таким чином, дилер повинен завжди мати актуальну інформацію про здійснені транзакції, обмеження та завершені позиції. Бухгалтеру потрібна інформація про попередні банківські дні та дані транзакцій, дійсний робочий день. Адміністратор повинен мати доступ до всіх даних без винятку.

Рис. 2.1. Корпоративнна мережа банку



Незважаючи на різноманітність організаційних типів у комерційних банках, їх структура, як правило, структурована навколо відділів front-офісу, бухгалтерії та управління.

Процес прийняття рішень у кожному підрозділі обмежений конкретними часовими рамками, починаючи від секунд до місяців або кварталів. Кожен рівень управління потребує якісно іншої інформації для прийняття рішень, яка

має певну релевантність. Крім того, для прийняття рішень підрозділам потрібні чіткі та різноманітні типи інформації.

Строки виконання прийнятого рішення залежать від рівня та типу. Кожен сектор комерційної банківської установи має власний підхід до прийняття та виконання рішень, деталі процесу управління банком диктують, що архітектура масштабованої інформаційної системи повинна відповідати наступним положенням щодо швидкості, для полегшення прийняття оперативних рішень призначте блок онлайн обмежень.

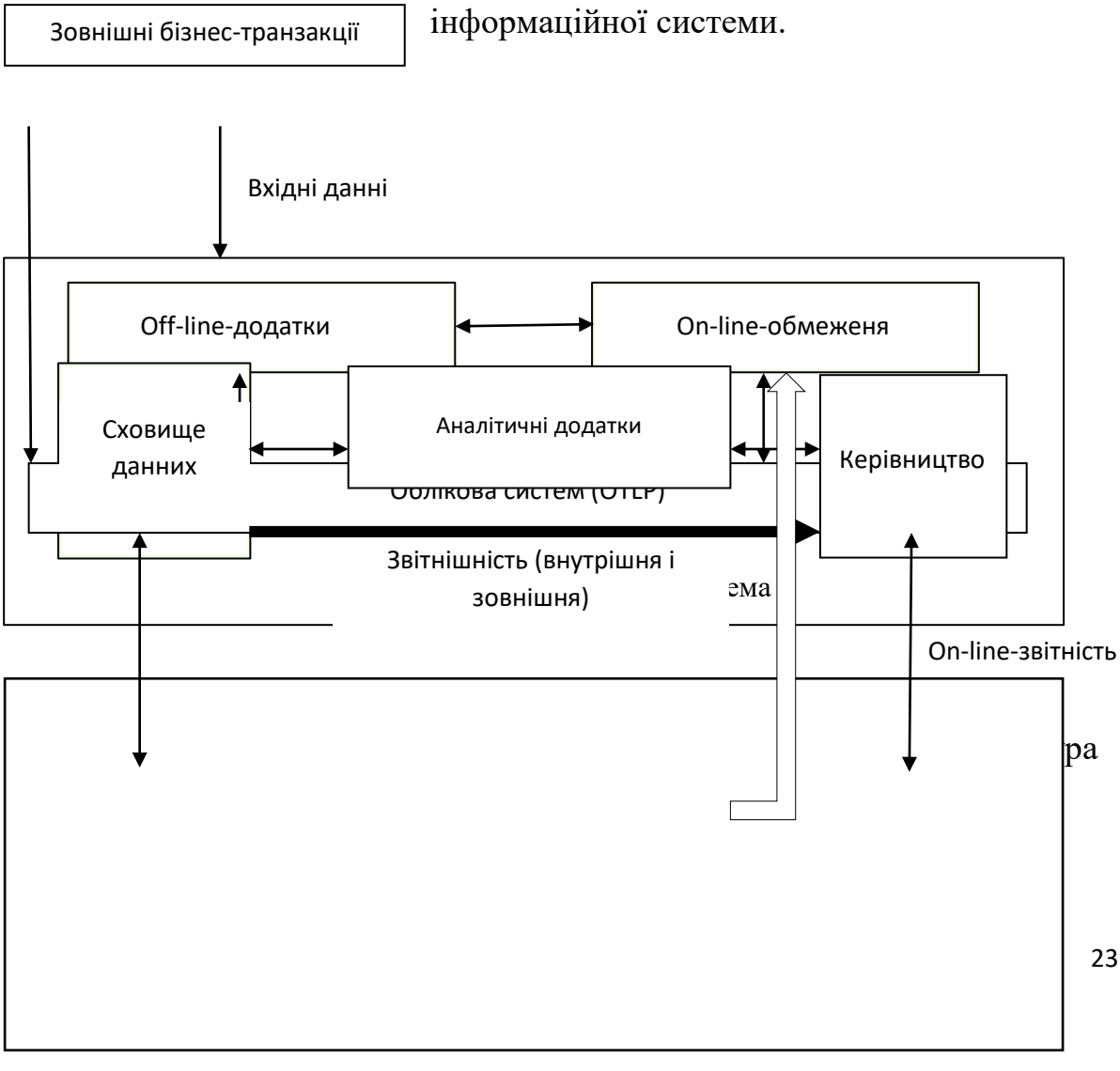
Відрегулювання розподілу баз даних і функцій системи обліку, звітності та аналітики відповідно до тактичних і стратегічних завдань управління.

Створення окремого підрозділу для найму менеджерів.

Використовувати як бухгалтерські, так і аналітичні навички.

Етичні методи.

Для забезпечення управління поточною діяльністю банку необхідно проаналізувати структуру його інформаційної системи.



За допомогою фронтальних додатків банк дає змогу здійснювати процес прийняття рішень на узгоджених умовах для клієнта та контрагентів. Бухгалтерський облік: фактичні дані, передбачені угодами, вводяться в облікову базу даних, яка забезпечує функції для back-офісу та рахунків. Традиційно інформаційною системою банку була база даних бухгалтерського обліку. Далі йде інформація, яка потрапляє в сховище даних і збирає всі показники.

Ці обмеження в першу чергу походять від системи бухгалтерського обліку, в якій зберігаються всі угоди та особисті рахунки. Як джерела може бути використана інформація щодо договорів, отримана з front-офісів, результати аналітичних заявок, а також прямі рішення керівництва банку.

Ключовою особливістю архітектури є перерозподіл функцій між аналітичною та обліковою підсистемами. Завдяки тісному зв'язку з аналітикою всі складні звіти подаються в аналітичну систему і становлять значну її частину. Ця система забезпечує більш складні засоби прийняття рішень, надаючи аналітичні можливості в поєднанні зі складними звітами. Усі звіти, необхідні Національному банку та внутрішньому контролю, можна отримати за допомогою цієї облікової системи.

Аналітичні програми використовуються для підготовки та оптимізації складних звітів, які надсилаються керівникам різних рівнів і Національному банку. Процес прийняття управлінських рішень отримує поточну онлайн-звітність з бази даних OLTP для всіх учасників.

Завдяки здатності сховища даних представляти та інтерпретувати звіти потужним способом, підрозділ є високоефективним. Рішення застосувати як OLTP, так і OPAP з величезними обсягами інформації також зумовлено високою вартістю.

У системі мають бути запроваджені спеціальні робочі простори для керівників, які поєднують облікові та аналітичні дані, необхідні для прийняття рішень». Аналізуючи роботу своїх підрозділів, керівники банку можуть формувати управлінські впливи на бізнес-процеси через ці панелі управління.

Керівники банку можуть використовувати зручні та ефективні інструменти, надані організацією, для прийняття своєчасних та ефективних рішень.

Система бухгалтерського обліку банку, яка базується на загальній базі даних, зазвичай об’єднує front-офіс, back-офіс і бухгалтерію. Саме ІБС виконує функції системи бухгалтерського обліку, займається питаннями розпорядження керівництва, керує кадровою роботою та контролює діяльність підрозділів. АРК містить програми для зберігання даних і аналітику.

Якщо архітектура ІБС дозволяє обробку даних на сервері, то надійність системи визначається тим, наскільки надійним є сервіс (за винятком питання про надійність каналів зв’язку). Вибрана СУБД повинна дозволяти розробку розподілених багаторівневих рішень з використанням серверів додатків і даних з індивідуальною структурою модуля додатків. Забезпечити точність і ефективність інформаційної системи зі значними даними за рахунок скорочення її витрат.

Схеми дублювання, які передбачають створення резервної бази даних, рекомендовані, оскільки вони дозволяють отримати негайний доступ до сервера-дублікату, щоб забезпечити його нормальну роботу після перерви. Коли дані та процедури дублюються та розповсюджуються із серверів, менша ймовірність спричинити тривалу перерву в роботі під час банківського бізнес-процесу. Це через це.

Рекомендується використовувати механізм бізнес-транзакцій, щоб прискорити відновлення системи та забезпечити виправлення помилок користувача на відміну від стандартних системних транзакцій, які охоплюють триваліші операції та підтримують більш точний запис подій програми (і системи) для документування всіх проміжних станів об'єкти господарювання до завершення операцій. У разі переривання господарської операції записи в журналі можуть вказати момент переривання та дозволити інформаційній системі банку відновити свою роботу в потрібний час. Логування необхідне для

забезпечення безпеки інформації в банках, оскільки дозволяє фіксувати всі дії користувача, в тому числі й невдалі.

Система бухгалтерського обліку є частиною інформаційної системи управління банком. Очевидно, що особливості процесу прийняття рішень вимагають наступного:

3. Інтегруйте кожну деталь, необхідну для прийняття управлінського рішення, у доступному для менеджера просторі.
4. Відповідальність за зовнішню інформацію та бізнес-процеси.
5. Адаптивність, що дозволяє оперативні управлінські маневри та модифікації для оптимізації функціонування комерційного банку в сучасних умовах».
6. Підтримання ефективності на позаштатних посадах можливо за допомогою надійності.
7. Продуктивність достатнього рівня для прийняття оперативних рішень. Через важливість інтеграції інформації в системі управління інформацією можна підтримувати декілька рівнів інтеграції даних одночасно.
8. Інтеграція базових даних на нижньому рівні. Основою для прийняття управлінських рішень є єдиний простір даних.
9. Повна інформація, яка зберігається в єдиній базі даних.
10. Діяльність фінансової установи, включаючи її філії.
11. Середній рівень інформаційних бізнес-процесів.
12. Банківська система на основі бухгалтерського обліку.
13. Очікується, що документообіг буде рівномірним. '.
14. Бухгалтерський облік, банківські продукти та клієнтські послуги присутні на рівнях.
15. Контрагенти, підрозділи. Основою для цього є єдина система довідників та індикаторів». Єдиний простір банківських бізнес-процесів пропонує користувачам можливість створювати прозорі технологічні ланцюжки взаємодії від кінця до кінця.

На верхньому поверсі розташовані автоматизовані робочі місця (АРМ). Взаємозв'язок між функціональними можливостями системи та інтерфейсом користувача досягається шляхом представлення всіх аспектів банківських бізнес-процесів в єдиному просторі. АРМ можуть звертатися до будь-якого функціонального аспекту інформаційної системи та її бізнес-процесів за допомогою доступу до системи адміністрування доступу, яка пронизує всі рівні.

Взаємодія банку та його відділень є формою інтеграції, і важливість цієї інтеграції важко переоцінити.

Загалом із філіями можна зв’язатися двома способами:Завдяки спільному простору, що містить функціональні точки, бізнес-процеси та дані, центральний офіс може здійснювати повний контроль над діями філії.

Використовуючи універсальну систему індикаторів, налаштованих на необхідний рівень деталізації, відділення банку можуть працювати автономно в автономному режимі та використовувати недорогі канали зв’язку. Це вигідно.

Система управління банком базується на багаторівневій системі, яка надає керівникам вичерпну інформацію про банк та його відділення та дозволяє ефективно управляти банківськими процесами.

Інформаційна система банку зазвичай взаємодіє з іншими методами автоматизації, включаючи комплексні системи «Клієнт банку», такі як банкомати, процесингові центри та технологію введення платіжних доручень із використанням сканованих даних. Керівництву банку необхідна інформація з цих джерел, і вкрай важливо розробити інформаційну систему, яка мінімізує втручання людини під час прийняття рішень.

Дуже важливо мати можливість інтегрувати ІБС із зовнішніми інструментами обробки інформації, такими як генератори звітів і аналітичні програми.

Зазвичай використовуються три основні методи передачі даних із зовнішніми програмами.

Протокол обміну враховує як онлайн, так і офлайн взаємодію.Онлайн-обмін побудований на основі стандартного протоколу TCP/IP, який не тільки полегшує обіг документів у реальному часі, але й дозволяє одночасно обробляти документи в кількох програмах. Найпростішим прикладом такої взаємодії є обробка платежу клієнта через зовнішню підсистему «Банк-клієнт» і отримання платіжного документа зі сканера.

 Офлайн-обмін призначений для отримання даних у пакетному режимі

Терміново і необхідно при використанні каналів зв'язку малої потужності.

 Через SQL Gateway. Це дозволяє зовнішнім програмам це робити.

Безпосередньо використовувати дані інформаційної системи за допомогою стандартного SQL. можна навести як приклад

Використання стандартних генераторів звітів і зовнішнього аналізу

Зручні програми та унікальні банківські програми

Взаємодіяти безпосередньо з системними даними.

 Дозволяє це на основі зовнішніх систем через об'єктний шлюз

Об’єкти для створення власних додатків. Основні поняття електронного документообігу

Банківські системи мають первинний механізми побудови бізнес-процесів який дозволяє своєчасно приймати рішення та здійснювати оперативні впливи. Документи подаються у front-офіс за домовленістю із замовником.

Торговий партнер «переміщується» через back-офіс або бухгалтерію. Прийняті рішення змінюють його зовнішній вигляд або перетворюють на інші документи. Рішення приймаються на рівнях front-офісу та деяких back-офісів

Напівавтоматичний (через поточні онлайн-обмеження). Рішення також можуть прийматися під час транзакції, які відображаються в певному наборі блок-схем (на основі результатів аналітичного застосування). Зміна виконання контракту шляхом прийняття рішень щодо документів та їх переходів між станами є одним із найважливіших методів управлінського впливу менеджерів. Іншою вимогою до ядра системи банківського контролю є гнучкість. Це означає:

можливість перегляду бізнес-програм за допомогою системи; чеки певних банків; один із способів здійснення управлінського впливу на банки в рамках

Оперативний рівень управління.Теорія про те, що «надійність банківської системи на першому місці» досі актуальна. Розглянемо основні фактори ризику та архітектурні рішення для їх нейтралізації.

- Поломка обладнання. В якості захисту використовується розподілена багаторівнева архітектура і системні транзакції, на основі яких будуються прикладні системи для бізнес-транзакцій.
- Помилки користувача та програмування. Заблоковано системою підтримки. Адаптація схем бізнес-процесів. У рамках функціональних пунктів системи ви можете налаштувати існуючі та створити нові каталоги та індикатори користувачів. Підсистема «Адміністратор банку» дозволяє створювати та редагувати плани документообігу, які утворюють функціональні модулі системи. Побудова організаційної структури. У просторі функціональних точок ви можете створити свою АРМ із будь-яким набором функцій. Це дуже важливо для підтримки оперативного прийняття рішень. Індивідуальна схема розподілу доступу до бізнес-процесів банку, його організаційної структури та об'єктів виробничого процесу в цілому забезпечує адаптацію системи до конкретної технології прийняття рішень банку в рамках існуючого набору функціональних моментів. Програмуючи та створюючи власні звіти, ви можете майже необмежено розширювати набір функціональних точок і властивостей системи.

"ІБС АРМ" - це набір функцій програми. Надаючи разом із системою набір інструментів розробки типу «Дизайнер екранної форми», ви можете програмувати власні функціональні частини системи з нуля, використовуючи системні функції або на основі наявних функцій. Створення звітів. Для системи потрібен вбудований генератор

Звіти, які мають доступ до даних усіх функцій. Крім того, використовуючи знання внутрішньої структури платформи ІБС, внутрішній генератор вигідно відрізняється від зовнішньої робочої швидкості.[6]

2.2 Аналіз способів несанкціонованого доступу та оцінка ризиків Інформаційної безпеки банку

Найбільш серйозною загрозою безпеці інформаційних ресурсів є витік або втрата таких ресурсів (особливо інформації, що становить банківську таємницю). Загрози джерелам інформації можуть бути реалізовані наступними способами::

- підкуп осіб, які мають прямий доступ до банківської таємниці та іншої інформації з обмеженим доступом до банківських установ;
- недбале, недбале поводження з банківською таємницею та іншою інформацією з обмеженим доступом;
- недотримання вимог щодо зберігання інформації з обмеженими правами доступу в банківських установах у контакті з регулюючими та наглядовими органами через відсутність юридичної та психологічної готовності відповідальних працівників, таких як банківські установи.

Протидія таким загрозам має складатися в першу чергу з:

- визначення надійності співробітників компаній, що займаються банківською таємницею та іншою інформацією з обмеженим доступом;
- організацію роботи приватного офісу з інформацією, компонентами та інформацією з обмеженим доступом до банківських установ;
- обмежений доступ. Тільки для того, щоб працівник міг виконувати покладені на нього функціональні завдання, він має обмежений доступ, з яким працівник може бути знайомий і виконувати певні дії.;
- особистий захист співробітника від інших засобів масової інформації, включаючи надані йому або розроблені ним документи, інформацію з обмеженим доступом до банківських установ.;
- обмежений доступ банківських установ обмежити доступ співробітників і неавторизованих осіб до об'єктів, в яких обробляється (зберігається) інформація;

- здійснення заходів з контролю за роботою співробітників, що використовують засоби масової інформації з обмеженим доступом до банківських установ, і виявлення протиправних дій з такою інформацією.

Впровадження надійних і ефективних систем зберігання носіїв інформації виключає несанкціоноване ознайомлення з ними, їх знищення або підробку.

До серйозних загроз безпеці інформаційної інфраструктури відносяться:

- неофіційний доступ і видалення захищеної інформації технічними засобами;
- запобігання поширенню інформації в засобах і системах зв'язку та комп'ютерних технологій, використання технічних засобів для розсекречення інформації, несанкціонований доступ до інформації та навмисний технічний вплив під час її обробки та зберігання;
- підслуховування відбуваються секретних переговорів з використанням технічних засобів.

Протидія таким загрозам засноване, перш за все, на широко поширеному і найбільш важливому з економічної точки зору використанні технічних засобів безпеки інформаційної інфраструктури.

Конкретні заходи щодо усунення загроз безпеці інформаційної інфраструктури фінансових установ включають:

- створення цілісності засобів захисту, технічного і програмного середовища, що полягає у фізичному збереженні засобів інформатизації, незмінності програмного середовища, виконанні засобами захисту передбачених функцій, ізольованості засобів захисту від користувачів;
- захист інформації від витоку внаслідок наявності фізичних полів за рахунок акустичних та побічних електромагнітних випромінювань і наводок на комунікаційні мережі та конструкції будівель;
- використання криптографічного захисту найбільш цінної інформації при її обробці в електронно-обчислювальних машинах (комп’ютерах), системах та комп’ютерних мережах і мережах електрозв’язку підприємства;

- надання диференційованого доступу працівникам для здійснення конкретних операцій (створення, читання, запис, модифікація, видалення) за допомогою програмно-технічних засобів, а також розмежування доступу користувачів до даних в електронно-обчислювальних машинах (комп'ютерах), системах та комп'ютерних мережах і мережах електрозв'язку банківської установи різного рівня та призначення;
- ідентифікація користувачів та здійснюваних ними процесів в електронно-обчислювальних машинах (комп'ютерах), системах та комп'ютерних мережах і мережах електрозв'язку установи на основі використання паролів, ключів, магнітних карт, цифрового підпису, а також біометричних характеристик особи як при доступі до інформаційно-телекомунікаційних систем;
- реєстрація (з фіксацією дати і часу) дій користувачів з інформаційними та програмними ресурсами в електронно-обчислювальних машинах (комп'ютерах), системах та комп'ютерних мережах, зокрема протиправних спроб доступу;
- попередження передачі інформації з обмеженим доступом по незахищених лініях зв'язку;
- запобігання впровадженню в інформаційно-телекомунікаційні системи програм-вірусів;
- регулярна перевірка технічних засобів і приміщень для виявлення наявності в них пристроїв несанкціонованого доступу до інформації;
- обладнання спеціальних приміщень для захисту мовної інформації при проведенні конфіденційних переговорів тощо.

Найбільшою загрозою безпеці в «інформаційній сфері» є погіршення ділового іміджу банківських установ, головним чином спричинене розповсюдженням інформації, у відносинах з реальними та потенційними клієнтами, конкурентами, керівництвом та правоохоронними органами. проблеми. Введення поки що недостовірної неправдивої та негативної

інформації про банківські установи впливає на її керівництво, працівників тощо.

Конкретними заходами щодо усунення загроз безпеці «інформаційного сектору» банківських установ є:

1. Оперативне реагування при поширенні неправдивої інформації про банківські установи.
2. Скоординоване та централізоване розповсюдження рекламної, маркетингової та іншої інформації, що покращує імідж та сприйняття банківських установ серед клієнтів.
3. Налагоджувати інформаційну співпрацю з органами державної влади та органами місцевого самоврядування в рамках чинного законодавства.

Виходячи з багатьох вищезазначених загроз та шляхів їх подолання, систему забезпечення інформаційної безпеки банківських установ можна визначити як комплекс організаційно-технічних, програмних та криптографічних заходів і протидії, таких як:

- захищати інформацію обмеженого доступу банківських установ від несанкціонованого поширення, використання та порушення конфіденційності (конфіденційності);

- забезпечення цілісності та доступності інформації, яка обробляється, зберігається та поширюється на електронно-обчислювальних машинах (ЕОМ), системах, комп'ютерних мережах та телекомунікаційних мережах банківських установ;

- протидія поширенню недостовірної та раніше неправдивої інформації про банківські установи та реалізації інформації негативного характеру, що негативно впливає на її діяльність.

2.3 Система управління інформаційною безпекою банківської установи та її документальне супроводження

Система управління інформаційною безпекою-це сучасний процес забезпечення безпеки інформаційних ресурсів організації, заснований на передовій міжнародній практиці. Стандарти Національного банку України засновані на міжнародних стандартах ISO27001 і ISO27002 і додають вимоги до захисту інформації у зв'язку з особливими потребами банківського сектора і нормативними вимогами, які в даний час представлені в нормативних документах Національного банку України. Законодавчо-нормативне регулювання захисту інформаційних ресурсів банківських установ представлено нормами Закону України «Про банки і банківську діяльність» і Закону України «Про інформацію». Окрім того в питаннях інформаційної безпеки банківських установ важливе місце займають і вимоги нормативно-правових актів НБУ з питань організації та управління інформаційною безпекою. Так, з метою підвищення рівня інформаційної безпеки установ банківської сфери з 2010 р. в Україні (відповідно до Постанови НБУ №474 від 28.10.2010р.) діють стандарти НБУ: СОУ Н НБУ 65.1 СУІБ 1.0:2010 «Методи захисту в банківській діяльності Система управління інформаційною безпекою. Вимоги» (ISO / IES 27001:2005, MOD); СОУ Н НБУ 65.1 СУІБ 2.0:2010 «Методи захисту в банківській діяльності. Звід правил для управління інформаційною безпекою» (ISO / IES 27002:2005, MOD).

Важливим документом є також «Методичні рекомендації щодо впровадження системи управління інформаційною безпекою та методики оцінки ризиків відповідно до стандартів Національного банку України від 01.03.2011р.». Ці методичні рекомендації розроблені на основі міжнародного стандарту ISO/IEC 27003:2010 з урахуванням особливостей банківської діяльності, стандартів та вимог Національного банку України з питань інформаційної безпеки.[5]

Як зазначає у своїй роботі на сьогодні в питаннях інформаційної безпеки прийняті, зокрема, такі міжнародні стандарти:

– серія ISO 27000 «Міжнародні стандарти для системи управління інформаційною безпекою»: ISO/IEC 27000:2009. Визначення і основні

принципи; ISO/IEC 27001:2005. Інформаційні технології – Методики безпеки – Системи менеджменту інформаційної безпеки – Вимоги (BS 7799-2:2005);

ISO/IEC 27002:2005. Інформаційні технології – Методики безпеки – Практичні правила управління інформаційною безпекою; ISO/IEC 27003:2010. Настанова з впровадження системи управління інформаційною безпекою; ISO/IEC 27005:2008. Інформаційні технології – Методика безпеки – Управління ризиками інформаційної безпеки (на основі стандарту BS 7799-3:2006); ISO/IEC 27006:2007. Інформаційні технології – Методики безпеки – Вимоги до організації, що провадять аудит і сертифікацію систем менеджменту інформаційної безпеки; ISO/IEC 27011:2008. Керівництво з менеджменту інформаційної безпеки для телекомунікацій; ISO/IEC 15408. Загальні критерії оцінки безпеки інформаційних технологій;

– серія ISO 13335 «Міжнародні стандарти безпеки інформаційних технологій»: ISO 13335-1:2004. Інформаційні технології – Керівництво по управлінню IT безпекою – Концепції і моделі для управління безпекою інформаційних і телекомунікаційних технологій; ISO 13335-3:1998.

Інформаційні технології – Керівництво по управлінню IT безпекою – Методи управління IT безпекою; ISO 13335-4:2000. Інформаційні технології – Керівництво по управлінню IT безпекою – Вибір механізмів захисту; ISO 13335-5:2001. Інформаційні технології – Керівництво по управлінню IT безпекою – Керівництво по управлінню мережевою безпекою.

Необхідність впровадження стандартів управління інформаційною безпекою в українських банках визначається вимогами Комітету Базель II щодо управління та мінімізації операційних ризиків у банках.

Впровадження стандартів управління інформаційною безпекою в українських банках дозволить:

- оптимізацію витрат на створення та підтримку систем захисту інформації.
- постійне відстеження та оцінку ризиків для бізнес-цілей.
- ефективне виявлення найбільш суттєвих ризики та зниження ймовірності їх реалізації.

- розробка ефективної політики інформаційної безпеки та забезпечити її якісне впровадження.
- ефективна розробка, впровадження та тестування планів відновлення бізнесу.
- переконатися, що керівництво банку та всі співробітники розуміють питання інформаційної безпеки.
- забезпечити підвищення репутації та ринкової привабливості банку.
- зменшення ризику пограбувань та інших нападів, які завдають шкоди банкам.

Слід зазначити, що зазначені вище переваги досягаються не лише завдяки «формальному» підходу до розробки, впровадження та функціонування систем управління інформаційною безпекою. Діяльність банку безпосередньо визначається зацікавленістю керівництва та працівників банку у підвищенні рівня інформаційної безпеки.

Крім того, впровадження стандартів управління інформаційною безпекою не є разовою подією. Фактично це безперервний процес розробки, впровадження, функціонування, моніторингу, перегляду, підтримки та вдосконалення СУІБ. Отже, методологічною основою управління інформаційною безпекою згідно стандартів серії ISO 27000 є процесний підхід.

Для ефективного управління організацією необхідно визначити та керувати різними видами діяльності. Будь-яка діяльність, яка споживає ресурси та керується для перетворення входів у виходи, може вважатися процесом. У багатьох випадках вихід одного процесу стає входом наступного процесу. Застосування систем процесів в організації та ідентифікація цих процесів, їх взаємодії та управління ними можна вважати «процесним підходом».

Процесний підхід до управління інформаційною безпекою наголошує на важливості:

а) зрозуміти вимоги організації до інформаційної безпеки та необхідність розробки політики та цілей інформаційної безпеки.

b) впровадити та забезпечити функціональність заходів безпеки для вирішення ризиків інформаційної безпеки організації в контексті загальних бізнес-ризиків організації.

c) моніторинг та перегляд продуктивності та ефективності СУІБ та постійного вдосконалення на основі об’єктивних вимірювань.

У рамках цього підходу для процесу СУІБ використовується модель «План-Виконання-Перевірка-Дія». Про це йдеться в огляді стандарту СОУ Н НБУ 65.1 СУІБ 1.0:2010. СУІБ використовує вимоги інформаційної безпеки та очікування зацікавлених сторін як вхідні дані та використовує необхідні заходи та процеси для створення вихідних даних інформаційної безпеки, які відповідають цим вимогам та очікуванням.

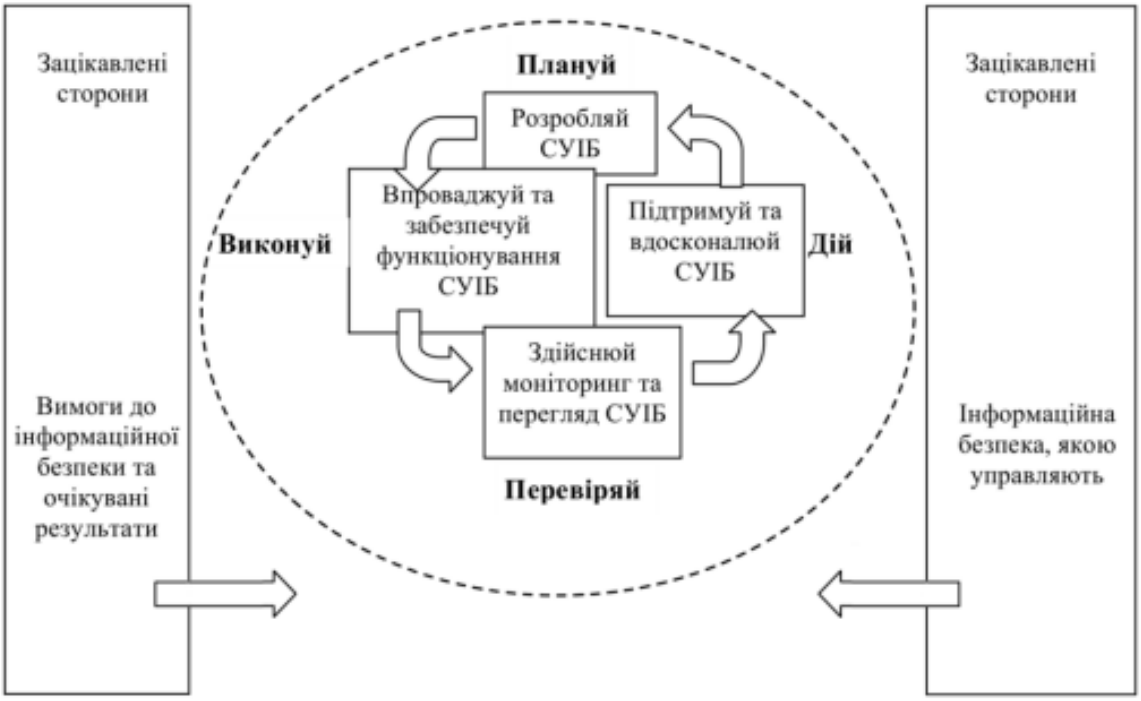


Рис.2.3 Система управлуння інформаційною безпекою

Безпека автоматизованих систем обробки інформації банку – властивість, що полягає у спроможності протидіяти спробам завдання збитків власникам і користувачам системи, тобто захищеності від спроб розкрадання чи руйнування її компонентів.

Головними завданнями будь-якої системи інформаційної безпеки є:

– забезпечення доступності даних для авторизованих користувачів – можливості оперативного отримання інформаційних послуг;

– гарантія цілісності інформації – її актуальності і захищеності від несанкціонованих змін або знищення;

– забезпечення конфіденційності відомостей.

Незважаючи на безліч можливостей витоку інформації, безпеку банківських даних та їх конфіденційність забезпечити цілком можливо.

2.4. Висновки до другого розділу

Корпоративна мережа банківських установ в основному має складну структуру, що об'єднує від 1 офісу до десятків, а іноді і сотень офісів і філій.

Безпека корпоративної мережі банку безпосередньо залежить від безпеки кожної мережі. Отримавши доступ до локальної мережі філій організації, зловмисник може поширювати шкідливе програмне забезпечення, таке як черв'яки, і отримати доступ до корпоративної мережі протягом декількох днів.

Зовнішні межі локальної мережі - це набір функцій безпеки, включаючи брандмауер, що відокремлює локальну мережу від загальнодоступного незахищеного Інтернету, і систему виявлення і запобігання вторгнень, яка реєструє і блокує всі несанкціоновані підключення із загальнодоступної мережі в локальну мережу. Він надійно захищений за допомогою захисного обладнання. Аналізуючи безпеку системи з використанням мінімально необхідного рівня програмної та апаратної захисту, виявляється безліч невирішених проблем.Основним недоліком системи є те, що вона не може своєчасно виявити факт проникнення в локальну мережу, оскільки антивірусні та антиспамові системи не дають 100% гарантії. Сюди входить той факт, що ви не можете цього зробити.

Системи мережевої безпеки розкидані по всьому світу, тому відсутність єдиної системи збору інформації може призвести до того, що факт злому системи може з'явитися занадто пізно або взагалі не з'явитися.

Без використання додаткових інструментів для аналізу поведінки користувачів і компонентів системи фахівці не зможуть розпізнати загрозу на початковому етапі симптомів.

РОЗДІЛ 3.ЗАСТОСУВАННЯ ДОДАТКОВИХ ПРОГРАМНИХ ЗАСОБІВ ЗАХИСТУ БАНКІВСЬКОЇ МЕРЕЖІ

3.1 Криптографічні способи захисту інформації

Існує досить велика кількість способів захисту комп'ютерів. Є методи, які ґрунтуються на застосуванні безпечних операційних систем та апаратного забезпечення, що здатне захистити комп'ютерну систему. Хоча під час проектування комп'ютерної системи необхідно взяти до уваги чимало характеристик. Безпека є серед них однією з найважливіших. Небезпечні програми деколи не правильно уподібнюються з комп'ютерними вірусами, тоді коли вірус – лише один із злочинних видів шкідливих програм.

В автоматизованих банківських системах (АБС) вибір засобів захисту інформації – досить складна задача, а при її рішенні особливо необхідно врахувати можливість різних протиправних дій щодо порушення працездатності такої системи, вартість реалізації засобів захисту і наявність різних зацікавлених сторін. Варто зазначити, що важливість забезпечення інформаційної безпеки оцінена і на державному рівні, що відбивається у вимогах нормативно-правових актів. Наприкінці 2017 року, Національний банк України встановив вимоги до кіберзахисту, які повинні впроваджуватися банками. Вимоги спрямовані на посилення захисту інформації у банківській системі з урахуванням актуальних кіберзагроз.

Заходи безпеки інформації включають:

1. Контроль доступу до ресурсів АБС (управління доступом).
2. Ідентифікація і автентифікація АБС (користувачів процесів і т.д.).
3. Реєстрація та аналіз подій, що відбуваються в АБС.
4. Контроль цілісності об’єктів АБС.
5. Шифрування даних.
6. Резервування ресурсів і компонентів АБС.

Кожен напрямок включає кілька етапів роботи. Управління доступу – захист інформації шляхом регулювання доступу до всіх ресурсів системи.

Регламентуються порядок роботи користувачів і персоналу, право доступу до окремих файлів в базах даних і т.д.

Доступ до даних банку захищається за допомогою системи ідентифікації, тобто паролями або електронними ключами. Ідентифікація – це присвоєння коду кожному об'єкту персонального ідентифікатора. Автентифікація – встановлення автентичності. Нові можливості дозволяють використовувати багатофакторну посилену ідентифікацію при авторизації в банківській системі. Така автентифікація особливо актуальна в роботі співробітників, що маю ть права введення і підтвердження фінансових документів.

Для аналізу ефективності вжитих заходів необхідно вести облік або запис, які будуть відзначати працездатність й дієвість застосованих засобів захисту інформації в банку. Ці функції забезпечують отримання й аналізінформації про стан ресурсів системи, реєстрацію дій, які можуть бути визначені як небезпечні ситуації, ведення журналу, який допоможе оперативно зафіксувати події, що відбуваються в системі. Аналіз журналу, якщо його вести належним чином, може допомогти у визначенні засобів, які використовував зловмисник під час порушення системи захисту, у визначенні реального стану системи, у виборі способів розслідування в разі порушення і підказати шляхи виправлення ситуації.

Контроль за цілісністю (захист від несанкціонованої модифікації суб'єктів системи) – контроль за цілісністю атрибутів суб'єкта, контроль за послідовністю і повнотою процесів та режимів їх виконання. Механізм контролю цілісності здійснює стеження за незмінністю контрольованих об'єктів, захист від шкідливого коду. При несанкціонованому знищенні, додаванні зайвих елементів та модифікації даних, зміну порядку розташування даних, формуванні фальсифікованих платіжних документів у відповідь на законні запити, активної ретрансляції повідомлень з їх затримкою. Цілісність порушується при, викраденні або незаконної зміні алгоритмів роботи. Забезпечення цілісності – частина комплексу заходів по досягненню безпеки інформації. Загрози, що відносяться до можливостей несанкціонованої

модифікації інформації, є загрозами цілісності. Загрози, що відносяться до можливостей несанкціонованого ознайомлення з інформацією є загрозами конфіденційності. В загальному випадку вважається, що для захисту інформації повинні бути створені механізми захисту. Це управління доступом до ресурсів, включаючи доступ до паролів, надання рівнів доступу до об’єктів, ідентифікація, реєстрація та облік роботи користувачів. Порушення цілісності може статись в наслідок наступних причин:

1. Помилки користувачів, які викликають викривлення чи втрату інформації.
2. Навмисні дії осіб, які не мають прав доступу до системи.
3. Збої обладнання, які викликають викривлення чи втрату інформації.
4. Фізичний вплив на носії інформації.
5. Вірусні впливи.

Одним з дієвих методів реалізації вимог цілісності інформації є крипто-графічний захист інформації (шифрування, хешування, електронний цифровий підпис).

При комплексному підході до захисту АБС, напрям забезпечення цілісності та доступності інформації переростає в план заходів, що спрямовані на забезпечення безперервності роботи АБС. Система шифрування даних забезпечує безпеку при обміні інформацією, тому всі дані, передані в банк або прийняті від банку, шифруються спеціальним методом згідно стандартів ISO 8730 та ISO 8731. Засоби шифрування доволі надійно захищають комп'ютерну інформацію від кіберзагроз. Кодування тексту за допомогою складних математичних алгоритмів, отримує все більшу популярність. Звичайно, що не один з алгоритмів шифрування не дає стовідсоткової гарантії захисту від зловмисників, але все ж, деякі методи шифрування досить складні, щоб дати змогу ознайомитися з повідомленнями зашифрованого змісту. Досить дієвим та потужним є застосування для захисту інформації криптозахисту, тобто систем, які дозволяють зашифрувати та дешифрувати інформаційні потоки.

RSA (абревіатура від англ. Прізвищ Rivest, Shamir та Adleman) – це один із поширених методів шифрування на сьогодні. Алгоритм, в основі якого кожен учасник процесу має власний таємний ключ та відкритий ключ, який не має бути секретним, за допомогою нього проводиться обмін повідомленнями. Електронний цифровий підпис (ЕЦП) – це дані в електронній формі, отримані за результатами криптографічного перетворення, які додаються до інших даних або документів і забезпечують їх цілісність та ідентифікацію автора. Криптографічні методи широко застосовуються у АБС та мають реалізацію у вигляді програмних, апаратних чи програмно-апаратних методів захисту інформації. Криптографія є провідним засобом забезпечення конфіденційності і контролю цілісності інформації.[7]

Безпека RSA залежить від обчислювальної складності розкладання великих цілих чисел на прості множники, тому міцність шифрування безпосередньо пов'язана з розміром ключа. Зі збільшенням обчислювальної потужності та відкриттям ефективніших алгоритмів розкладання на множники — зростає й здатність розкладати на множники все більші й більші числа.

Алгоритм RSA складається з чотирьох етапів:

1. Генерація ключа. Даний етап, своєю чергою, ділиться ще на декілька підетапів:

- вибір двох простих чисел. На даному етапі необхідно обрати два великі прості числа p та q (просте число — це таке, що може ділитись без залишку лише на 1 та на себе). Числа p і q повинні триматись в таємниці, оскільки саме на них базується процес створення приватного та публічного ключа. Для більшої надійності p та q повинні: бути обрані навмання; бути великими; мати велику різницю.
- визначення модуля. На даному етапі необхідно визначити число n, що використовується як модуль для відкритого і закритого ключа: $n = p * q$. Довжина n, що виражена в бітах, і є довжиною ключа.
- застосування функції Ейлера. Далі необхідно визначити значення функції Ейлера від числа n, що має наступний вигляд: $\varphi(n) = (p — 1) * (q — 1)$.

- вибір відкритої експоненти. Після визначення значення функції Ейлера необхідно випадково обрати ціле число e таке, що 2 < e < φ(n). Число e (яке також називають відкритою експонентою) повинно бути взаємно простим до значення φ(n). Занадто малі значення відкритої експоненти можуть послабити алгоритм RSA.
- знаходження приватної експоненти (оберненого за модулем числа). В той час, коли відкрита експонента e є частиною публічного ключа, то приватна експонента d (або ж секретна) являється частиною приватного ключа. Приватна експонента d знаходиться як обернене за модулем φ(n) до числа e, тобто: d * e ≡ 1(mod(φ(n)). Значення оберненого за модулем числа можна визначити з допомогою розширеного алгоритму Евкліда.

2. Розподіл ключа. Тепер, коли визначено всі необхідні числа, можна сформувати приватний та публічний ключ. Приватний ключ складається з пари d та n, а публічний з пари e та n.Число d повинно триматись у таємниці, оскільки воно використовується для дешифрування повідомлення. Числа p, q та φ(n) також повинні триматись у таємниці, адже з допомогою них можна визначити приватну експоненту d, але після її визначення ці числа можна одразу відкинути. Публічний ключ пересилається з допомогою надійного, але не обов'язково зашифрованого каналу зв'язку.

3. Шифрування. Для того, щоб зашифрувати текст m, необхідно обчислити таку рівність: c = me (mod n), де c — зашифрований текст, m — простий текст.

4. Дешифрування. Для того, щоб розшифрувати текст c, необхідно обчислити таку рівність: m = cd (mod n).

Приклад. Зашифруємо повідомлення КНИГА, що складається із символів українського алфавіту та представляється як послідовність цілих чисел
M = 1417 10 3 0. Для простоти обчислень будемо використовувати невеликі числа, проте пам'ятаємо, що на практиці застосовують дуже великі прості числа. Оберемо $p = 3$ і $q = 11$, тоді $n = p \cdot q = 3 \cdot 11 = 33$.

Обчислимо $\varphi(33) = 2 \cdot 10 = 20$.

Виберемо (випадково) $e = 3$ та перевіримо виконання умов: 1 < 3 < 20,

НСД(3, 20) =1.

Визначимо d – ключ дешифрування з рівняння $3d \equiv 1(mod\ 20)$.

Для розв'язання рівняння використаємо розширений алгоритм Евкліда:

1) послідовно виконуємо ділення з остачею попереднього значення $ri{-}1$ на наступне ri , у відповідністю з рівністю $ri{-}1 = riqi{+}1 + ri{+}i$ (якщо $ri = 1$, тоді зупиняєио процес);

2) використовуємо рекурентне співвідношення $ui{+}1 = ui{-}1 - qi{+}1ui$;

3) використовуємо рекурентне співвідношення $vi{+}1 = vi{-}1 - qi{+}1vi$;

4) щоб почати процес виконання алгоритму, використовуємо значення $r0 = 20$, $r1 = 3$, $u0 = 1$, $u1 = 0$, $v0 = 0$, $v1 = 1$.

Виконуємо перевірку $3 \cdot 7\ mod\ 20 = 1$. Таким чином, $d = 7$.

Опублікуємо відкритий ключ $(e, n) = (3, 33)$.

i	r_i	q_i	u_i	v_i
0	20		1	0
1	3		0	1
2	20 mod 3 =2	0 div 3 = 6	1-6·0=1	0-6·1=-6
3	3 mod 2 =1	3 div 2 = 1	0-1·1=-1	1-1·(-6)=7

Зберігаємо в таємниці секретний ключ $(d, n) = (7, 33)$.

Зашифруємо повідомлення M = 14 17 10 3 0, що складається із п'яти блоків mi:

$c_1 = 14^3\ mod\ 33 = ((14^2\ mod\ 33) \cdot (14^1\ mod\ 33))\ mod\ 33 = (31 \cdot 14)\ mod\ 33 = 434\ mod\ 33 = 5$;

$c_2 = 17^3\ mod\ 33 = ((17^2\ mod\ 33) \cdot (17^1\ mod\ 33))\ mod\ 33 = (25 \cdot 17)\ mod\ 33 =$
$= 425\ mod\ 33 = 29$;

$c_3 = 10^3\ mod\ 33 = 1000\ mod\ 33 = 10$;

$c_4 = 3^3\ mod\ 33 = 27\ mod\ 33 = 27$;

$c_5 = 0^3\ mod\ 33 = 0\ mod\ 33 = 0$.

Шифротекст: C = 5 29 10 27 0.

Для дешифрування потрібно також виконати піднесення до степеня, використовуючи ключ дешифрування 7:

$m_1 = 5^7 \ mod\ 33 = ((5^4 \ mod\ 33) \cdot (5^3 \ mod\ 33)) \ mod\ 33 = (31 \cdot 26) \ mod\ 33 = 806 \ mod\ 33 = 14$;

$m_2 = 29^7 \ mod\ 33 = ((29^4 \ mod\ 33) \cdot (29^3 \ mod\ 33)) \ mod\ 33 = (((29^2))^2 \ mod\ 33) \cdot (29^2 \ mod\ 33) \cdot (29 \ mod\ 33)) \ mod\ 33 = (25 \cdot 16 \cdot 29) \ mod\ 33 = 11600 \ mod\ 33 = 17$;

$m_3 = 10^7 \ mod\ 33 = ((10^4 \ mod\ 33) \cdot (10^3 \ mod\ 33)) \ mod\ 33 = (((10^2))^2 \ mod\ 33) \cdot (10^2 \ mod\ 33) \cdot (10 \ mod\ 33)) \ mod\ 33 = (1 \cdot 1 \cdot 10) \ mod\ 33 = 10 \ mod\ 33 = 10$;

$m_4 = 27^7 \ mod\ 33 = ((27^4 \ mod\ 33) \cdot (27^3 \ mod\ 33)) \ mod\ 33 == (((27^2))^2 \ mod\ 33) \cdot (27^2 \ mod\ 33) \cdot (27 \ mod\ 33)) \ mod\ 33 = (9 \cdot 3 \cdot 27) \ mod\ 33 = 729 \ mod\ 33 = 3$;

$m_5 = 0^7 \ mod\ 33 = 0 \ mod\ 33 = 0$.

Відкритий текст: M = 14 17 10 3 0 - КНИГА.

Алгоритм Діффі-Геллмана (англ. Diffie–Hellman key exchange (D–H)) — це метод обміну криптографічними ключами. Один з перших практичних прикладів обміну ключами, що дозволяє двом учасникам, що не мають жодних попередніх даних один про одного, отримати спільний секретний ключ із використанням незахищеного каналу зв'язку. Цей ключ можна використати для шифрування наступних сеансів зв'язку, що використовують шифр з симетричним ключем.

Хоча протокол Діффі-Геллмана є анонімним (без автентифікації) протоколом встановлення ключа, він забезпечує базу для різноманітних протоколів з автентифікацією, і використовується для забезпечення цілковитої прямої секретності в недовговічних режимах Transport Layer Security (відомих як EDH або DHE залежно від комплектації шифру).

Алгоритм Діффі — Геллмана, де K — підсумковий спільний секрет (ключ)

Припустимо, обом абонентам відомі деякі два числа: велике просте p (наприклад, 600 цифр) $g \in \{1, ..., p\}$ (як варіант, вони можуть бути «зашиті» в програмне забезпечення), які не є секретними та можуть бути відомі й іншим

$$b \in \{1, ..., p-1\}$$

зацікавленим особам. Для того, щоб створити не відомий нікому іншому секретний ключ, обидва учасники генерують великі випадкові числа: перший — $a \in \{1, ..., p-1\}$, другий- .Потім перший учасник обчислює значення $A=g^a \bmod p$ і відправляє його другому, а той обчислює $B=g^b \bmod p$ і відправляє першому. Вважається, зловмисник може отримати ці повідомлення, але не змінити їх. (Рис.3.1.)

На другому етапі, перший користувач на основі свого a і отриманого мережею B обчислює значення $B^a \bmod p=g^{\{ab\}} b \bmod p$, а другий користувач з b і A обчислює значення $A^b \bmod p=g^{\{ab\}} \bmod p$. Як неважко бачити, в обох користувачів виходить те саме число: $K=g^{\{ab\}} \bmod p$. Його вони і можуть використовувати як секретний ключ, оскільки зловмисник тут зтикається з необхідністю обчислити функцію $DH_g(g^a, g^b) = g^{\{ab\}} \bmod p$.

Припустимо, користувачі A і B мають намір обмінятися ключами за

алгоритмом Діффі-Хелмана, суть якого полягає в наступному:

1. A і B спільно обирають просте число p і ціле число g таке, що $1 < g < p - 1$

і g є первісним коренем p.

Первісним коренем за модулем p називається таке число g, що при піднесення

до степеню g

$imod\ p$ всі його степені $i \in \{1, \ldots, p-1\}$ за модулем p пробігають

по всім числам взаємно простим із p.

Нехай $p = 5$. Усі взаємно прості числа з p: 1, 2, 3, 4.

Елементи 2 та 3 є первісними коренями 5.

1
$1^1 mod\ 5 = 1$
$1^2 mod\ 5 = 1$
$1^3 mod\ 5 = 1$
$1^4 mod\ 5 = 1$
2
$2^1 mod\ 5 = 2\ mod\ 5 = 2$
$2^2 mod\ 5 = 4\ mod\ 5 = 4$

$2^3 mod\ 5 = 8\ mod\ 5 = 3$
$2^4 mod\ 5 = 16\ mod\ 5 = 1$
3
$3^1 mod\ 5 = 3\ \text{mod}\ 5 = 3$
$3^2 mod\ 5 = 9\ \text{mod}\ 5 = 4$
$3^3 mod\ 5 = 27\ \text{mod}\ 5 = 2$
$3^4 mod\ 5 = 81\ \text{mod}\ 5 = 1$
4
$4^1 mod\ 5 = 4\ \text{mod}\ 5 = 4$
$4^2 mod\ 5 = 16\ \text{mod}\ 5 = 1$
$4^3 mod\ 5 = 64\ \text{mod}\ 5 = 4$
$4^4 mod\ 5 = 256\ \text{mod}\ 5 = 1$

2. Користувач А вибирає випадкове ціле число $x < p$, обчислює $x_A = g^x\ modp$

та відправляє його користувачеві В.

3. Користувач В вибирає випадкове ціле число $y < p$, обчислює $y_B = g^y modp$

та відправляє його користувачеві А.

4. Користувач А обчислює закритий ключ за формулою $k_A = y\ {}^x_B mod\ p$.

5. Користувач В обчислює закритий ключ за формулою $k_B = x_A^y\ mod\ p$.

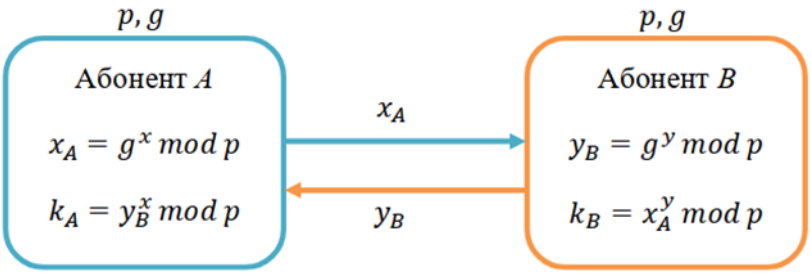


Рис.3.1. Алгоритм Діффі — Геллмана

Ці дві формули обчислення дають однакові результати. Відкритими параметрами є: p, g, x_A та y_B. Закриті параметри: x, y

Суворий облік каналів та серверів, а також заходи, що забезпечують технічний захист інформації і безпеку банку мають на увазі захистрезервних копій, забезпечення безперебійного живлення устаткування, що містить цінну інформацію, обмежений доступ до сейфів та захист від витоку інформації акустичним способом.

Резервування ресурсів та абонентів АБС передбачає: організацію трегулярних процедур порятунку і резервного зберігання критичних даних, періодичну перевірку резервних пристроїв обробки даних, підготовку фахівців, здатних замінити адміністраторів систем, реєстрацію систем та зберігання носіїв інформації в суворо визначених місцях, видачу їх уповноваженим особам з необхідними відмітками в реєстр траційних документах.

Безпека банкоматів та платіжних терміналів повинна забезпечуватися з використанням традиційних засобів – антивірусного захисту. В той же час специфіка таких пристроїв вимагає застосування додаткових засобів захисту. Створення «замкнутого програмно-апаратного середовища», повністю виключає установку любого стороннього програмного забезпечення і підключення зовнішніх пристроїв.

Система безпеки в цілому це безперервний процес ідентифікації, аналізу та контролю. Оскільки інформація, що знаходиться в базі даних банків являє собою реальну матеріальну цінність, то вимоги до зберігання та обробки цієї інформації завжди будуть підвищеними.

Уточнення і доповнення безлічі актуальних загроз безпеки банківської інформації, безпека інформації і кібербезпека в банківському секторі, це основа для створення нового синергетичного підходу в області інформаційної безпеки АБС. Для аналізу основних видів загроз безпеки банківської інформації використовується відома модель безпеки – тріада CIA (Confidentiality, Integrity, Availability).(Рис.3.2)

У моделі «конфіденційність» – забезпечення доступу до інформації тільки авторизованим користувачам, «цілісність» – забезпечення достовірності і повноти інформації, «доступність» – забезпечення доступу до інформації.

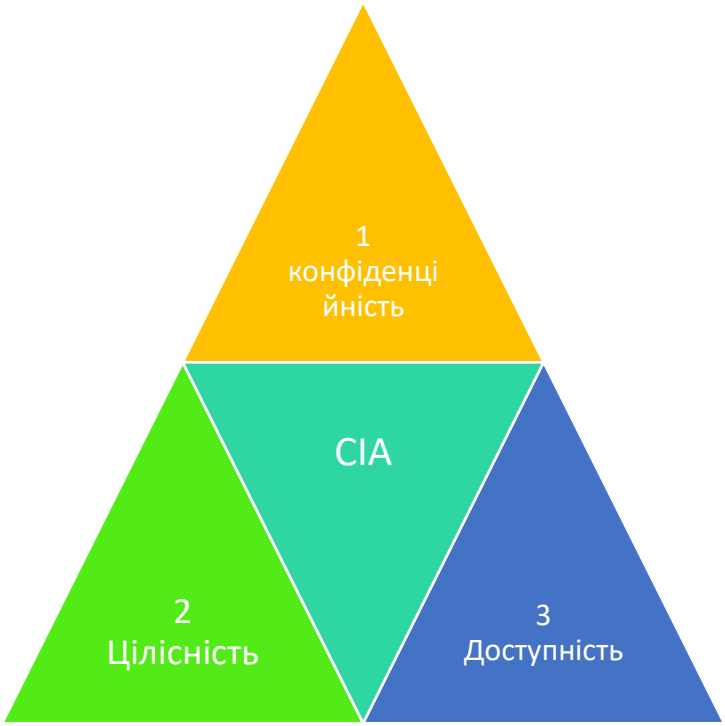


Рис.3.2. тріада CIA (Confidentiality, Integrity, Availability).

Модель синергетичного підходу – оцінка безпеки банківських систем. В процесі аналізу ризиків інформаційної безпеки можуть використовуватися спеціалізовані програмні комплекси, що дозволяють автоматизувати процес аналізу вихідних даних та розрахунку значень ризику.

Метою інформаційної безпеки є забезпечення трьох найважливіших сервісів безпеки. Відповідно моделі безпеки інформації включають: конфіденційність, цілісність і доступність. Слід зазначити ключову особливість, характерну тільки пропонованому синергетичному підходу до безпеки банківської інформації. Основна мета запропонованого підходу – це порушення в системі забезпечення банківської інформації керованих емерджентних властивостей, спрямованих на отримання синергетичного ефекту, який досягається завдяки якісно новому підходу до безпеки. Таким чином, виходячи із потреби дотримання правила триєдиної позиції до забезпечення безпеки банківської інформації в рамках синергетичного підходу при взаємодії вибраних профілів безпеки і з метою підвищення рівня її захищеності є оцінювання величини ризику аналогічного грошового капіталу.

3.2 Застосування серверу Nginx

Nginx це спеціальне ПЗ з відкритим кодом, сумісне з UNIX-системами. Розроблявся як веб-сервер для обслуговування запитів HTTP. Над проектом працював програміст Ігор Сисоєв. Розробка почалася в 2002 році, реліз ПЗ – грудень 2004 року. Основною метою було вирішення проблеми C10k, пов'язаної зі складністю обробки багаточисленних запитів (10000 запитів і більше). Створений веб-сервер успішно справлявся з високими навантаженнями, чим і зумовлено його подальшу популярність, незважаючи на існування серйозного конкурента в особі веб сервера Apache.[16]

На даний момент веб-сервер Nginx є одним з найнадійніших серверних програм завдяки успішній реалізації керованої подіями та асинхронної архітектури. Його використовують такі інтернет-гіганти, як Google, WordPress, Netflix та багато інших. Найбільш поширені варіанти застосування:

1. В якості самостійного HTTP-сервера;
2. У вигляді SMTP, IMAP, POP3-сервера;
3. У зв'язці з Apache, де Nginx відведена роль сервера, що кешує;

Для встановлення та перевірки наявності Nginx можна виконати дві прості команди:

Apt-get install Nginx

Nginx -v

В результаті виконання останньої команди буде відображено відомості про версію встановленого програмного забезпечення.

Основний Nginx є асинхронні алгоритми, що не блокують event-driven. Сервер генерує робочі процеси, завдяки чому йому вдається одночасно обробляти величезну кількість запитів, що надходять. Усі робочі процеси виконуються незалежно один від одного. У межах кожного робочого процесу виконуються робочі з'єднання. Обробка з'єднань відбувається лише в тому випадку, якщо було згенеровано нову подію.

Таким чином, маємо тришарову архітектуру веб-сервера:

Робочі з'єднання – структурні одиниці робочого процесу.

Запити надходять від робочих з'єднань до робочих процесів.

Всі дані надходять до головного (основного) процесу, що надає кінцеві результати обробки.

Алгоритми, що використовуються, сприяють високій масштабованості системи навіть на відносно слабких машинах. Nginx - однопотоковий сервер, що не генерує процеси для кожного нового з'єднання. Цим обумовлена рівномірність використання ресурсів фізичного сервера (процесор, ОЗП) навіть під час обробки великої кількості запитів.

Основні переваги, порівняння з Apache. Доцільність вибору того чи іншого рішення для веб-сервера визначається на основі таких критеріїв:

Продуктивність. Nginx показує високу швидкість обробки підключень статичного контенту. За цим показником він обходить найближчого конкурента (Apache) вдвічі. Продуктивність при роботі з динамічними сайтами обох програмних продуктів приблизно однакова.

Використання ресурсів. Nginx є менш вимогливим до пам'яті, ніж веб-сервер Apache.

Сумісність із ОС. Nginx підтримує багато популярних операційних систем. Однак він розроблявся для UNIX-систем. Сумісність з Windows реалізована слабко, тому швидкість роботи програмного забезпечення в цій системі досить низька.

Підтримка користувача. Допомога клієнтам надається у рамках e-mail-листування. Існують також форуми спільнот, на яких обговорюють різні питання.

Основні відмінності від Apache:

Швидка обробка запитів, пов'язаних із статичним контентом. Однак Nginx не містить алгоритмів для самостійної обробки запитів до динамічних даних. Для цього використовується зовнішній процесор, який виконує функції обробки та повертає підсумковий результат Nginx. Останній, своєю чергою, відправляє його клієнту.

Заборона на можливість перевизначення конфігураційних файлів на рівні директорій, що зумовлює приріст продуктивності та безпеки в порівнянні з Apache, що інтерпретує кожен .htaccess-файл.

Орієнтир на роботу з URI в першу чергу, які лише за потреби транслюються у запити до файлової системи. Такий підхід забезпечує поєднання двох основних функціональностей: проксі-сервер та веб-сервер.

Відсутність механізму динамічного підключення різних модулів (для шифрування, проксіювання, поштових функцій та інші). З цим пов'язані як переваги (безпека, підключення тільки необхідних модулів), так і недоліки (необхідність ручного складання, нижча гнучкість рішення порівняно з Apache).

Висока масштабованість при більш низьких вимогах до обчислювальних систем (фізичних серверів).

Розглянуті характеристики веб-серверів, зумовлюють вибір на користь того чи іншого кожного конкретного проекту. У деяких випадках доцільно використовувати зв'язок Apache + Nginx. Останній розгортається перед Apache для виконання функцій реверс-проксі. За обробку всіх запитів відповідає Nginx, здатний успішно справлятися з їх великою кількістю. Його основне завдання у цій конфігурації – обробка статичного контенту. Якщо потрібно виконання, наприклад, PHP-сценаріїв, запит надходить на Apache, де відбувається його обробка. Отриманий результат передається спочатку Nginx,а потім – до кінцевого користувача.

Таким чином, Nginx сортує запити на статичні та динамічні. З першими він успішно справляється сам, інші – ndash; адресує Apache. Цей підхід спричиняє часткове зниження навантаження на останній.

Пов'язування Apache + Nginx використовується для горизонтального масштабування веб-додатків. Наприклад, можливий варіант підключення кількох веб-серверів Apache до одного Nginx, що розподіляє навантаження між ними. Підхід сприяє підвищенню стійкості до відмови веб-сервісів.

Багато сервісів FREEhost.UA, наприклад віртуальний хостинг, також використовують зв'язку Apache + Nginx. Apache відповідає за роботу з динамічним контентом, а Nginx – за статичний контент. Оскільки веб-сервер Nginx знаходиться попереду, проксируючи весь трафік на Apache, за його допомогою ми фільтруємо частину “шкідливого трафіку” під час DDOS атаки та спроб злому сайтів.

3.3. Системи управління інформаційною безпекою (SIEM, SOAR)

З кожним днем зростає складність і кількість різних загроз інформаційної безпеки. Разом з цим збільшується і число систем, покликаних захистити бізнес від цих загроз. У 99% великих компаній функціонує міжмережевий екран, антивірусне рішення і система виявлення вторгнення — це сьогодні необхідний мінімум. Крім того, в мережі працюють бази даних, операційні системи та програмне забезпечення власної розробки. Всі ці підсистеми генерують реєстраційні журнали і різні події. А якщо компанія має кілька філій або віддалених офісів, то потік даних від інформаційних підсистем збільшується в десятки разів.[14]

У підсумку адміністратори отримують сотні тисяч повідомлень від безлічі різноманітних підсистем кожен день. Функціонування кожної з підсистем окремо критично для бізнесу в цілому, тому фахівці змушені аналізувати весь цей потік інформації. Виділити важливі повідомлення стає все складніше, і в результаті цінність окремих рішень для забезпечення безпеки прагне до нуля, а час відновлення інформаційної системи після збоїв катастрофічно зростає.

Максимально ефективно використовувати дані, одержані від сенсорів (серверів) виявлення атак і від міжмережевих екранів атаках (про відображених ними атаках) дозволяє використання системи моніторингу інформаційної безпеки. Система моніторингу ІБ дозволяє звести всі події та інциденти ІБ в єдиній консолі, виконує інтелектуальний аналіз атак та їх наслідків і допомагає адміністраторам виробити контрзаходи. Крім цього, система моніторингу ІБ виконує реєстрацію та зберігання всіх подій інформаційної безпеки, що робить

можливим використання отриманого матеріалу в якості доказового при виконанні розслідувань інцидентів та судочинстві.

Основні можливості SIEM-систем:

- Збір інформації про події з різних пристроїв забезпечення інформаційної безпеки і мережевих пристроїв;
- Візуалізацію подій в режимі реального часу;
- Підтримку сигнатурних і «поведінкових» методів виявлення аномалій і атак;
- Можливість створення власних правил кореляції;
- Можливість управління активними мережевими пристроями з метою блокування шкідливого трафіку;
- Прогнозування результатів атаки;
- Аналіз ризику захищаємої системи;
- Автоматичне визначення статусу події (атака, сканування тощо);
- Можливість обробки та аналізу інцидентів безпеки;
- Фокусування уваги на пріоритетних захищаємих вузлах;
- Вбудована система роботи з інцидентами, можливість інтеграції з існуючою;
- Автоматична реакція на інциденти.
- забезпечити централізоване управління подіями і інцидентами ІБ
- збільшити швидкість виявлення, розслідування та реагування на інциденти
- управляти інцидентами ІБ
- підвищити ефективність управління ризиками ІБ
- підвищити рівень відповідності політикам і нормативним вимогам

SIEM або система управління інформацією та подіями безпеки, є фундаментальним елементом для забезпечення кібербезпеки. Програмне забезпечення SIEM дозволяє використовувати утиліти, необхідні для ефективного управління журналами, виявляє вторгнення, кореляцію подій, збір інформації про загрози, управління інцидентами, виконання стандартів

відповідей та оцінки вразливості. Звичайно, різні інструменти SIEM будуть віддавати пріоритет варіативним функціям. Важливо, щоб користувач зрозумів основи SIEM, перш ніж вибрати інструмент, який він хоче використовувати. Незалежно від того, вирішить він встановити безкоштовну або платну програму SIEM, слід звернути увагу на наступні важливі фактори: виявлення вторгнень: ефективний підхід до виявлення вторгнень має вирішальне значення. Інст румент повинен відрізняти нешкідливі невдалі спроби входу від інтенсивних, симптоматичних атак. Ключовим моментом є аналіз даних у реальном часі; автоматичні інформування та оповіщення: SIEM рішення повинно попереджувати користувача про виникнення будь-яких проблем; ведення журналу подій допомагає виявити незвичайну активність у режимі реального часу та ретельно її дослідити; інтелектуальне виявлення загроз: SIEM рішення повинно бути здатним прогнозувати потенційні загрози, що вимагає порівняння інформації про останні загрози з поточними; зберігання та фільтрація даних: дані повинні зберігатися в архіві, щоб при необхідності на них можна було посилатися, інформуючи про подальші виявлення загроз. Ці дані повинні бути доступні для пошуку та фільтрації для того, щоб користувачі могли легко і швидко орієнтуватися; візуалізація даних може бути надзвичайно корисна для їх інтерпретації. Графіки, лічильники та кольорове кодування мають змогу миттєво надати користувачу представлення про те, що відбувається в системі; відповідні вимоги: завжди корисно мати програмне забезпечення SIEM, яке надає змогу забезпечити надання необхідних нормативних вимог; сумісність: програмне забезпечення SIEM має бути сумісним із наявною системою, оскільки тільки так воно дозволить користувачам мати всебічне представлення про поточні події.

Оркестрування, автоматизація та реагування на загрози безпеки (SOAR) - це набір програм, розроблених для посилення кібербезпеки організації. Платформа SOAR дозволяє команді аналітиків з безпеки відстежувати дані про безпеку з різних джерел, включаючи системи інформації та управління безпекою і платформи розвідки загроз. [13]

Використовуючи платформу SOAR, команда безпеки може підвищити ефективність і скоротити час реагування. Вона збирає інформацію про загрози, автоматизує рутинні реакції і сортує більш складні загрози, мінімізуючи необхідність втручання людини.

Рішення SOAR визначають пріоритети та стандартизують дії з реагування на інциденти, щоб команди безпеки могли співпрацювати в розслідуванні та управлінні інцидентами. Робочі процеси, які можна автоматизувати, проходять через стандартизовані процеси реагування, визначені в плейбуках.

Платформи SOAR відрізняються в залежності від постачальника, але всі вони повинні включати ці ключові функції:

Оркестрування: Рішення SOAR може полегшити зв'язок між інструментами безпеки і продуктивності, такими як фаєрволи і засоби виявлення вторгнень.

Автоматизація: Рішення SOAR може автоматизувати стандартні робочі процеси з кібербезпеки, такі як виявлення сигналів безпеки та можливих вторгнень.

Реагування: Платформа SOAR може працювати як з автоматизованими, так і з ручними процесами для підтримки своєчасного реагування на загрози безпеці.

Інтеграція: Платформа SOAR може працювати з різноманітними додатковими продуктами безпеки для підтримки загальної системи безпеки організації.

Команди безпеки регулярно стикаються з великою кількістю загроз, таких як шкідливе програмне забезпечення та фішинг.

Автоматизація кібербезпеки є ключем до управління цим постійним потоком загроз. Платформи машинного навчання можуть покращити реагування на інциденти, навчаючись на історичних даних і діючи незалежно, щоб людські ресурси могли виконувати завдання, які неможливо автоматизувати.

Інструменти SOAR також можуть покращити реагування на інциденти, передбачаючи загрози до того, як вони відбудуться. Зі збільшенням кількості розумних пристроїв в мережі збільшується і кількість точок входу для хакерів.

Банківські установи, використовують системи SOAR для асиміляції даних з цих окремих пристроїв і швидкого реагування на потенційні загрози безпеці до того, як зловмисники зможуть їх реалізувати. Це допомагає їм досягти кіберстійкості.

SOAR допомагає командам безпеки використовувати зібрані дані для оптимізації операцій за допомогою автоматизації безпеки та використання сценаріїв.

Пріоритезація загроз: SOAR допомагає командам безпеки визначати пріоритети та групувати сповіщення для більш ефективного виявлення та розслідування загроз.

Звітність та аналіз: Платформи SOAR можуть генерувати звіти, які допомагають командам безпеки виявляти тенденції в організації.

Інформаційна панель безпеки: Платформи SOAR можуть слугувати центральною інформаційною панеллю, яка допомагає командам безпеки відстежувати та спільно реагувати на тривоги.

Перш ніж розглядати рішення SOAR, важливо проаналізувати загальну систему безпеки установи. Спочатку організація повинна мати надійну систему безпеки зі стандартизованими сценаріями дій і бібліотекою робочих процесів реагування.

Коли ваші операції з безпеки будуть повністю розроблені, ви можете зосередитися на автоматизації встановлених процесів безпеки за допомогою передового інструменту безпеки, такого як SOAR.

Рішення SOAR та SIEM відіграють різні ролі у ваших операціях з безпеки. Єдиною метою програмного рішення SIEM є збір та надсилання сповіщень співробітникам служби безпеки для розслідування.

Інструмент SOAR використовує дані про проблеми безпеки для автоматизації реагування. SOAR також використовує штучний інтелект для прогнозування та реагування на подібні загрози в майбутньому.

Співробітники служби безпеки часто використовують як інструмент SOAR, так і інструмент SIEM. Ці дві платформи доповнюють одна одну і можуть працювати разом для забезпечення вашої загальної безпеки.

Взаємозв'язок між ними нагадує відносини помічника з менеджером. Рішення SIEM збирає і співвідносить журнали, щоб визначити ті, які відповідають критеріям оповіщення. Воно має архів логів і можливості аналізу, які не вбудовані в SOAR-платформи.

Коли ви використовуєте платформу SOAR з платформою SIEM, SOAR може отримувати дані від SIEM, а потім виконувати резолюції. SOAR слугує центром для команд безпеки, де вони можуть отримати контекст і відреагувати на оповіщення.

Без SOAR командам безпеки доведеться використовувати різноманітні інтерфейси за межами SIEM. З SOAR і SIEM разом команди безпеки можуть працювати ефективно, покладаючись на платформи, які показують їм, які оповіщення потребують подальшого розслідування і вирішення.

Подібно до того, як команди безпеки можуть отримати вигоду від використання SIEM з SOAR, інші продукти безпеки можуть використовувати можливості вашого рішення SOAR. Наприклад, платформа розвідки загроз для розширення можливостей розслідування загроз рішення SOAR.

3.3 Система поведінкового аналізу користувачів (UEBA)

Аналіз поведінки користувачів та суб'єктів (UEBA) – це підхід до кібербезпеки, який базується на аналізі поведінки користувачів та об'єктів в системах організації. Ця інноваційна технологія дозволяє виявляти аномальну активність, яка може вказувати на можливі загрози безпеці. UEBA базується на

принципі вивчення нормальної поведінки користувачів та ресурсів, а потім виявляє будь-які відхилення від цих норм.

Ви стикаєтеся з постійним шквалом загроз, про деякі з яких ви навіть не підозрюєте. Реальність така, що ваші користувачі стоять за багатьма загрозами та порушеннями, як зловмисними, так і випадковими. Як типова точка входу для атаки, користувачі є складним об'єктом для моніторингу та захисту. Щоб протистояти хвилі атак, вам потрібно зосередити свою увагу на користувачах, використовуючи можливості аналітики поведінки користувачів та організацій (UEBA).

UEBA – це рішення для кібербезпеки, яке застосовує аналітику для відстеження поведінки користувачів і організацій та виявлення потенційної несанкціонованої активності, яка може свідчити про кібератаку. [15]

Рішення UEBA надає центрам управління безпекою (SOC) видимість для виявлення загроз з боку користувачів, які в іншому випадку могли б залишитися невиявленими, а також можливість захисту від різних атак різного рівня складності. У цьому можуть допомогти ефективні інструменти безпеки UEBA:

a) Обробляти машинні дані в схему, що відповідає вимогам безпеки
b) Отримати справжнє уявлення про користувачів, а не лише про розрізнені облікові записи
c) Виявляти та визначати пріоритети складних загроз, пов'язаних з користувачами
d) Прискорити кваліфікацію та розслідування загроз
e) Оптимізувати реагування через робочі процеси операцій з безпеки

Оцінюючи інструменти безпеки UEBA, важливо пам'ятати про основні сценарії використання, виходячи з конкретних потреб і вимог вашої організації. На високому рівні безпека UEBA може допомогти вам виявити та відреагувати на такі випадки використання UEBA: компрометація облікового запису, інсайдерська загроза, а також зловживання та неправомірне використання привілейованих облікових записів.

Ваша організація збирає та генерує надзвичайну кількість даних з різних джерел. Перш ніж аналізувати ці дані, їх потрібно нормалізувати та збагатити, щоб уможливити ефективний пошук та машинний аналіз. Без успішної підготовки даних для аналізу ваше рішення UEBA неминуче міститиме "сліпі зони", створюючи помилкові спрацьовування, пропускаючи важливі дії, або, що ще гірше, створюючи помилкові спрацьовування, неправильно характеризуючи нешкідливі аномалії як загрози.

Обробка даних починається з розбору машинних даних на поля метаданих, спеціально структуровані для аналітики безпеки. Застосування єдиної схеми до оброблюваних даних є ключовим моментом для UEBA. При уважному розгляді можна виявити значну різницю між потужністю цих можливостей у різних рішеннях. Наприклад, при повідомленні про зміну дозволів іншого користувача адміністратором, схема повинна бути здатна відрізнити адміністратора від користувача, на якого це вплинуло. Нормалізація даних підвищує точність аналізованих даних шляхом коригування значень на основі відомих відхилень.

Збагачення даних містить процес додавання метаданих, отриманих з логу, з додатковими контекстними даними для більш ефективного аналізу. Нижче наведено кілька прикладів збагачення даних.

Використання геолокації для перетворення IP-адреси на передбачуване місцезнаходження

Декодування кодів логу в змістовну та діагностичну класифікацію постачальників (наприклад, Windows Event ID 4624 = успішний вхід в обліковий запис).

Класифікація даних особливо цінна для ефективного аналізу різноманітного обладнання та постачальників (наприклад, розуміння загальних значень числових кодів Check Point, Cisco і Palo Alto). Також корисно розуміти загальні дії, які може використовувати аналітика, наприклад, всі автентифікації, тип автентифікації, місцезнаходження і час використання облікового запису, незалежно від базової інфраструктури.

Загалом, UEBA може надавати комплексну картину подій і дій, що відбуваються в організації, допомагаючи вчасно виявляти та протидіяти загрозам. Завдяки UEBA, компанії можуть підвищити ефективність своїх заходів з кібербезпеки та забезпечити високий рівень захисту від сучасних кіберзагроз.

3.4 Висовки до третього розділу

Створення системи моніторингу та управління інформаційною безпекою має важливе значення при розробці складних засобів захисту мережі як програмного, так і апаратного забезпечення. Управління кожною системою окремо є надзвичайно складним і може бути недосяжним у певних ситуаціях. За допомогою SIEM ви можете отримувати.

Інформує про поточний стан компонентів локальної мережі в режимі реального часу. Запис і зберігання звітів, які надають точні оцінки безпеки системи.

Система аналізу поведінки може бути окремою програмою або доповненням до системи SIEM. Унікальною особливістю сучасних систем, створених з використанням штучного інтелекту та машинного навчання, є аналіз складних систем для виявлення нетипової активності в сегментах мережі.

Експерти з інформаційної безпеки в банку використовують систему автоматизації для реагування на інциденти інформаційної безпеки, що спрощує процес, допомагаючи їм визначити, чи не загрожує їм зашкодити їхній локальній мережі. Реалізація таких заходів значно посилить захист локальної мережі банківської установи та дозволить завчасно ідентифікувати та запобігати потенційним загрозам і атакам, коли зловмиснику не вдається проникнути за зовнішній периметр мережі.

ВИСНОВКИ

У цьому дослідженні ми дослідили систему захисту системи банку, можливі загрози, ризики, оцінку ризиків та системи.

Ми проаналізували загрози інформаційній безпеці банків, тому, як ми бачимо, в структурі захисту даних банківських установ визначена основна складова - безпека ресурсів та інфраструктури.

Майже четверта частина банківських установ використовують інноваційні інструменти кібербезпеки, і майже половина опитаних фінансових установ обмежені загальними інструментами захисту своєї мережевої інфраструктури.

На жаль, ці статистичні дані підтверджуються чисельними атаками на банківський сектор, які призводять до значних фінансових і репутаційних втрат, а в деяких випадках і до повного припинення існування банківських установ.

При аналізі безпеки були виявлені деякі недоліки її захисту:

1. Антивірус не може гарантувати, що він повністю захистить вашу систему від шкідливих програм;

2. Програмне забезпечення не можна використовувати для виявлення вразливостей нульового дня

Або апаратні засоби захисту;

3. Незаконна поведінка працівників;

4. Відсутність єдиного контролю над всією системою;

5. Повільний час відгуку на події інформаційної безпеки;

6. Можливість неповного усунення інцидентів інформаційної безпеки;

Доцільність впровадження додаткового програмного забезпечення для забезпечення безпеки та контролю в мережу місцевих фінансових установ обговорювалася наступним чином::

- криптографічні засоби захисту інформації
- застосування серверу Nginx
- системи управління інформаційною безпекою (SIEM, SOAR)
- система поведінкового аналізу користувачів (UEBA)

Комплексне впровадження цих засобів безпеки дозволяє відстежувати функціональність всіх елементів локальної мережі організації в режимі реального часу, збирати і аналізувати дані про всі програмних і апаратних компонентах мережі, аналізувати поведінку систем і співробітників організації, а також аналізувати встановлені моделі поведінки. При виявленні будь-яких відхилень від мережі відділ інформаційної безпеки банку буде негайно повідомлений про можливу кібератаку на локальну мережу.

Як тільки працівник підтверджує загрозу конфіденційності, доступності та цілісності інформації в локальній мережі, система автоматизації реагування на інциденти інформаційної безпеки робить все необхідне для усунення загрози та мінімізації втрат Компанії, негайно дотримуючись директив, визначених у разі інциденту. Це значно прискорює згубність реагування на інформаційні інциденти та усуває помилкову поведінку співробітників, пов'язану з "людськими факторами"."

Список використаних джерел

1. Абрамова А. С. Система ризиків діяльності комерційних банків в умовах цифровізації. Проблеми і перспективи економіки та управління. 2021. № 4(28). С. 186-193.

2. Трофіменко О. Г., Прокоп Ю. В., Логінова Н. І., Задерейко О. В. Кібербезпека України: аналіз сучасногостану. Захист інформації. 2019. Т. 21, № 3. С. 150-157.

3.. Forcadell F. J., Aracil E., & Úbeda, F. The Impact of Corporate Sustainability and Digitalization on InternationalBanks' Performance. Global Policy. 2020. № 11 (S1). P. 18-27. https://onlinelibrary.wiley.com/doi/10.1111/1758-5899.12761

4..Національний банк України. URL: https://bank.gov.ua/

5. Про затвердження Положення про організацію кіберзахисту в банківській системі України та внесеннязмін до Положення про визначення об'єктів критичної інфраструктури в банківській системі України: постанова Правління Національного банку України від 12 серпня 2022 року № 178.

URL: https://bank.gov.ua/ua/legislation/Resolution_12082022_178

6..Курченко О. Б.. Інформаційні системи і технології в банківських та фінансо-вих установах: Навч. посіб. — К.: МАУП, 2006. — 224 с.: іл. — Бібліогр.: с. 218–219.

7.Усік П.С., Буравченко К.О. Безпека банківських систем : навч. посіб. / П. С. Усік, К. О. Буравченко; М-во освіти і науки України, Центральноукр. нац. техн. ун-т.— Кропивницький: ЦНТУ, 2022. — 194 с.

8.Постанова «Про внесення змін до деякиNo 40х нормативно-правових актів Національного банку України» від 30 березня 2023

9.Черевко О. В. Джерела виникнення загроз інформаційній безпеці банківських установ / О. В. Черевко, В. М. Андрієнко, І. Ю. Напора // Вісник Черкаського університету. Серія: Економічні науки. – 2016. – No 3. – С.120-127. ст. 21 Закону України "Про інформацію" [Електронний ресурс] – Режим доступу: https://zakon.rada.gov.ua/laws/show/2657-12#Text

10. ст. 60 Закону України "Про банки і банківську діяльність" [Електронний ресурс] - Режим доступу: https://zakon.rada.gov.ua/laws/show/2121-14#Text

11. ст. 200 Цивільного кодексу України [Електронний ресурс] - Режим доступу: https://zakon.rada.gov.ua/laws/show/435-15#Text

12. Постанова 28.09.2017 No 95 Про затвердження Положення про організаціюзаходів із забезпечення інформаційної безпеки в банківській системі України [Електронний ресурс] - Режим доступу:

https://zakon.rada.gov.ua/laws/show/v0095500-17#Text

13.Андрущенко К.Ю., Ющенко. АВТОМАТИЗАЦІЯ ПРОЦЕСУ РЕАГУВАННЯ НА ІНЦИДЕНТИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ОРГАНІЗАЦІЇ. Сучасний захист інформації No4(52) 2022-[Електронний ресурс] - Режим доступу: https://journals.dut.edu.ua/index.php/dataprotect/article/view/2666/2560

14. УПРАВЛІННЯ ІНФОРМАЦІЄЮ ТА ПОДІЯМИ БЕЗПЕКИ, SIEM, [Електронний ресурс] - Режим доступу: https://iitd.com.ua/upravlinnja-informaciieju-ta-podijami-bezpeki-siem/

15. Johansen G. Digital forensics and incident response: an intelligent way to respond to attacks. – 2017.

16. nginx- -[Електронний ресурс] - Режим доступу: https://nginx.org/

Додатки

Додаток А

```
@@ -519,6 +519,21 @@

     qc = ngx_quic_get_connection(c);

+    if (!ngx_quic_keys_available(qc->keys, ctx->level, 1)) {
+        ngx_log_error(NGX_LOG_ALERT, c->log, 0, "quic %s write keys discarded",
+                      ngx_quic_level_name(ctx->level));
+
+        while (!ngx_queue_empty(&ctx->frames)) {
+            q = ngx_queue_head(&ctx->frames);
+            ngx_queue_remove(q);
+
+            f = ngx_queue_data(q, ngx_quic_frame_t, queue);
+            ngx_quic_free_frame(c, f);
+        }
+
+        return 0;
+    }
+
     ngx_quic_init_packet(c, ctx, &pkt, qc->path);

     min_payload = ngx_quic_payload_size(&pkt, min);
```

Додаток Б

reusing crypto contexts for packet protection

```
@@ -335,6 +335,7 @@
    qc->validated = pkt->validated;

    if (ngx_quic_open_sockets(c, qc, pkt) != NGX_OK) {
+       ngx_quic_keys_cleanup(qc->keys);
        return NULL;
    }

@@ -585,6 +586,8 @@

    ngx_quic_close_sockets(c);

+   ngx_quic_keys_cleanup(qc->keys);
+
    ngx_log_debug0(NGX_LOG_DEBUG_EVENT, c->log, 0, "quic close completed");

    /* may be tested from SSL callback during SSL shutdown */

--- a/src/event/quic/ngx_event_quic_openssl_compat.c    Fri Oct 20 18:05:07 2023 +0400
+++ b/src/event/quic/ngx_event_quic_openssl_compat.c    Fri Oct 20 18:05:07 2023 +0400
@@ -54,9 +54,10 @@
```

```
 static void ngx_quic_compat_keylog_callback(const SSL *ssl, const char *line);
-static ngx_int_t ngx_quic_compat_set_encryption_secret(ngx_log_t *log,
+static ngx_int_t ngx_quic_compat_set_encryption_secret(ngx_connection_t *c,
     ngx_quic_compat_keys_t *keys, enum ssl_encryption_level_t level,
     const SSL_CIPHER *cipher, const uint8_t *secret, size_t secret_len);
+static void ngx_quic_compat_cleanup_encryption_secret(void *data);
 static int ngx_quic_compat_add_transport_params_callback(SSL *ssl,
     unsigned int ext_type, unsigned int context, const unsigned char **out,
     size_t *outlen, X509 *x, size_t chainidx, int *al, void *add_arg);
@@ -214,14 +215,14 @@
         com->method->set_read_secret((SSL *) ssl, level, cipher, secret, n);
         com->read_record = 0;

-        (void) ngx_quic_compat_set_encryption_secret(c->log, &com->keys, level,
+        (void) ngx_quic_compat_set_encryption_secret(c, &com->keys, level,
                                                      cipher, secret, n);
     }
 }


 static ngx_int_t
-ngx_quic_compat_set_encryption_secret(ngx_log_t *log,
+ngx_quic_compat_set_encryption_secret(ngx_connection_t *c,
     ngx_quic_compat_keys_t *keys, enum ssl_encryption_level_t level,
     const SSL_CIPHER *cipher, const uint8_t *secret, size_t secret_len)
 {
@@ -231,6 +232,7 @@
     ngx_quic_hkdf_t      seq[2];
     ngx_quic_secret_t   *peer_secret;
     ngx_quic_ciphers_t   ciphers;
+    ngx_pool_cleanup_t  *cln;

     peer_secret = &keys->secret;

@@ -239,12 +241,12 @@
     key_len = ngx_quic_ciphers(keys->cipher, &ciphers, level);

     if (key_len == NGX_ERROR) {
-        ngx_ssl_error(NGX_LOG_INFO, log, 0, "unexpected cipher");
+        ngx_ssl_error(NGX_LOG_INFO, c->log, 0, "unexpected cipher");
         return NGX_ERROR;
     }

     if (sizeof(peer_secret->secret.data) < secret_len) {
-        ngx_log_error(NGX_LOG_ALERT, log, 0,
+        ngx_log_error(NGX_LOG_ALERT, c->log, 0,
                       "unexpected secret len: %uz", secret_len);
         return NGX_ERROR;
     }
@@ -262,15 +264,43 @@
     ngx_quic_hkdf_set(&seq[1], "tls13 iv", &peer_secret->iv, &secret_str);

     for (i = 0; i < (sizeof(seq) / sizeof(seq[0])); i++) {
-        if (ngx_quic_hkdf_expand(&seq[i], ciphers.d, log) != NGX_OK) {
+        if (ngx_quic_hkdf_expand(&seq[i], ciphers.d, c->log) != NGX_OK) {
```

```
            return NGX_ERROR;
        }
    }

+    /* register cleanup handler once */
+
+    if (peer_secret->ctx) {
+        ngx_quic_crypto_cleanup(peer_secret);
+
+    } else {
+        cln = ngx_pool_cleanup_add(c->pool, 0);
+        if (cln == NULL) {
+            return NGX_ERROR;
+        }
+
+        cln->handler = ngx_quic_compat_cleanup_encryption_secret;
+        cln->data = peer_secret;
+    }
+
+    if (ngx_quic_crypto_init(ciphers.c, peer_secret, 1, c->log) == NGX_ERROR) {
+        return NGX_ERROR;
+    }
+
    return NGX_OK;
 }


+static void
+ngx_quic_compat_cleanup_encryption_secret(void *data)
+{
+    ngx_quic_secret_t *secret = data;
+
+    ngx_quic_crypto_cleanup(secret);
+}
+
+
 static int
 ngx_quic_compat_add_transport_params_callback(SSL *ssl, unsigned int ext_type,
    unsigned int context, const unsigned char **out, size_t *outlen, X509 *x,
@@ -578,8 +608,7 @@
    ngx_memcpy(nonce, secret->iv.data, secret->iv.len);
    ngx_quic_compute_nonce(nonce, sizeof(nonce), rec->number);

-    if (ngx_quic_crypto_seal(ciphers.c, secret, &out,
-                             nonce, &rec->payload, &ad, rec->log)
+    if (ngx_quic_crypto_seal(secret, &out, nonce, &rec->payload, &ad, rec->log)
        != NGX_OK)
    {
        return NGX_ERROR;

--- a/src/event/quic/ngx_event_quic_output.c    Fri Oct 20 18:05:07 2023 +0400
+++ b/src/event/quic/ngx_event_quic_output.c    Fri Oct 20 18:05:07 2023 +0400
@@ -941,13 +941,17 @@
    res.data = dst;
```

```
    if (ngx_quic_encrypt(&pkt, &res) != NGX_OK) {
+       ngx_quic_keys_cleanup(pkt.keys);
        return NGX_ERROR;
    }

    if (ngx_quic_send(c, res.data, res.len, c->sockaddr, c->socklen) < 0) {
+       ngx_quic_keys_cleanup(pkt.keys);
        return NGX_ERROR;
    }

+   ngx_quic_keys_cleanup(pkt.keys);
+
    return NGX_DONE;
 }


@@ -26,9 +26,8 @@
 static uint64_t ngx_quic_parse_pn(u_char **pos, ngx_int_t len, u_char *mask,
     uint64_t *largest_pn);

-static ngx_int_t ngx_quic_crypto_open(const ngx_quic_cipher_t *cipher,
-    ngx_quic_secret_t *s, ngx_str_t *out, u_char *nonce, ngx_str_t *in,
-    ngx_str_t *ad, ngx_log_t *log);
+static ngx_int_t ngx_quic_crypto_open(ngx_quic_secret_t *s, ngx_str_t *out,
+    u_char *nonce, ngx_str_t *in, ngx_str_t *ad, ngx_log_t *log);
 static ngx_int_t ngx_quic_crypto_hp(ngx_log_t *log, const EVP_CIPHER *cipher,
     ngx_quic_secret_t *s, u_char *out, u_char *in);

@@ -108,13 +107,14 @@
 ngx_quic_keys_set_initial_secret(ngx_quic_keys_t *keys, ngx_str_t *secret,
     ngx_log_t *log)
 {
-    size_t             is_len;
-    uint8_t            is[SHA256_DIGEST_LENGTH];
-    ngx_str_t          iss;
-    ngx_uint_t         i;
-    const EVP_MD       *digest;
-    ngx_quic_hkdf_t    seq[8];
-    ngx_quic_secret_t  *client, *server;
+    size_t              is_len;
+    uint8_t             is[SHA256_DIGEST_LENGTH];
+    ngx_str_t           iss;
+    ngx_uint_t          i;
+    const EVP_MD        *digest;
+    ngx_quic_hkdf_t     seq[8];
+    ngx_quic_secret_t   *client, *server;
+    ngx_quic_ciphers_t  ciphers;

     static const uint8_t salt[20] =
         "\x38\x76\x2c\xf7\xf5\x59\x34\xb3\x4d\x17"
@@ -180,7 +180,25 @@
         }
     }

+    if (ngx_quic_ciphers(0, &ciphers, ssl_encryption_initial) == NGX_ERROR) {
```

```
+        return NGX_ERROR;
+    }
+
+    if (ngx_quic_crypto_init(ciphers.c, client, 0, log) == NGX_ERROR) {
+        return NGX_ERROR;
+    }
+
+    if (ngx_quic_crypto_init(ciphers.c, server, 1, log) == NGX_ERROR) {
+        goto failed;
+    }
+
     return NGX_OK;
+
+failed:
+
+    ngx_quic_keys_cleanup(keys);
+
+    return NGX_ERROR;
 }


@@ -343,9 +361,9 @@
 }


-static ngx_int_t
-ngx_quic_crypto_open(const ngx_quic_cipher_t *cipher, ngx_quic_secret_t *s,
-    ngx_str_t *out, u_char *nonce, ngx_str_t *in, ngx_str_t *ad, ngx_log_t *log)
+ngx_int_t
+ngx_quic_crypto_init(const ngx_quic_cipher_t *cipher, ngx_quic_secret_t *s,
+    ngx_int_t enc, ngx_log_t *log)
 {

 #ifdef OPENSSL_IS_BORINGSSL
@@ -357,19 +375,7 @@
         ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_AEAD_CTX_new() failed");
         return NGX_ERROR;
     }
-
-    if (EVP_AEAD_CTX_open(ctx, out->data, &out->len, out->len, nonce, s->iv.len,
-                          in->data, in->len, ad->data, ad->len)
-        != 1)
-    {
-        EVP_AEAD_CTX_free(ctx);
-        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_AEAD_CTX_open() failed");
-        return NGX_ERROR;
-    }
-
-    EVP_AEAD_CTX_free(ctx);
 #else
-    int             len;
     EVP_CIPHER_CTX  *ctx;

     ctx = EVP_CIPHER_CTX_new();
@@ -378,114 +384,9 @@
```

```
        return NGX_ERROR;
    }

-   if (EVP_DecryptInit_ex(ctx, cipher, NULL, NULL, NULL) != 1) {
-       EVP_CIPHER_CTX_free(ctx);
-       ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_DecryptInit_ex() failed");
-       return NGX_ERROR;
-   }
-
-   in->len -= NGX_QUIC_TAG_LEN;
-
-   if (EVP_CIPHER_CTX_ctrl(ctx, EVP_CTRL_AEAD_SET_TAG, NGX_QUIC_TAG_LEN,
-                           in->data + in->len)
-       == 0)
-   {
-       EVP_CIPHER_CTX_free(ctx);
-       ngx_ssl_error(NGX_LOG_INFO, log, 0,
-                     "EVP_CIPHER_CTX_ctrl(EVP_CTRL_AEAD_SET_TAG) failed");
-       return NGX_ERROR;
-   }
-
-   if (EVP_CIPHER_CTX_ctrl(ctx, EVP_CTRL_AEAD_SET_IVLEN, s->iv.len, NULL)
-       == 0)
-   {
-       EVP_CIPHER_CTX_free(ctx);
-       ngx_ssl_error(NGX_LOG_INFO, log, 0,
-                     "EVP_CIPHER_CTX_ctrl(EVP_CTRL_AEAD_SET_IVLEN) failed");
-       return NGX_ERROR;
-   }
-
-   if (EVP_DecryptInit_ex(ctx, NULL, NULL, s->key.data, nonce) != 1) {
-       EVP_CIPHER_CTX_free(ctx);
-       ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_DecryptInit_ex() failed");
-       return NGX_ERROR;
-   }
-
-   if (EVP_CIPHER_mode(cipher) == EVP_CIPH_CCM_MODE
-       && EVP_DecryptUpdate(ctx, NULL, &len, NULL, in->len) != 1)
-   {
-       EVP_CIPHER_CTX_free(ctx);
-       ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_DecryptUpdate() failed");
-       return NGX_ERROR;
-   }
-
-   if (EVP_DecryptUpdate(ctx, NULL, &len, ad->data, ad->len) != 1) {
-       EVP_CIPHER_CTX_free(ctx);
-       ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_DecryptUpdate() failed");
-       return NGX_ERROR;
-   }
-
-   if (EVP_DecryptUpdate(ctx, out->data, &len, in->data, in->len) != 1) {
+   if (EVP_CipherInit_ex(ctx, cipher, NULL, NULL, NULL, enc) != 1) {
        EVP_CIPHER_CTX_free(ctx);
-       ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_DecryptUpdate() failed");
-       return NGX_ERROR;
```

```
-    }
-
-    out->len = len;
-
-    if (EVP_DecryptFinal_ex(ctx, out->data + out->len, &len) <= 0) {
-        EVP_CIPHER_CTX_free(ctx);
-        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_DecryptFinal_ex failed");
-        return NGX_ERROR;
-    }
-
-    out->len += len;
-
-    EVP_CIPHER_CTX_free(ctx);
-#endif
-
-    return NGX_OK;
-}
-
-
-ngx_int_t
-ngx_quic_crypto_seal(const ngx_quic_cipher_t *cipher, ngx_quic_secret_t *s,
-    ngx_str_t *out, u_char *nonce, ngx_str_t *in, ngx_str_t *ad, ngx_log_t *log)
-{
-
-#ifdef OPENSSL_IS_BORINGSSL
-    EVP_AEAD_CTX  *ctx;
-
-    ctx = EVP_AEAD_CTX_new(cipher, s->key.data, s->key.len,
-                           EVP_AEAD_DEFAULT_TAG_LENGTH);
-    if (ctx == NULL) {
-        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_AEAD_CTX_new() failed");
-        return NGX_ERROR;
-    }
-
-    if (EVP_AEAD_CTX_seal(ctx, out->data, &out->len, out->len, nonce, s->iv.len,
-                          in->data, in->len, ad->data, ad->len)
-        != 1)
-    {
-        EVP_AEAD_CTX_free(ctx);
-        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_AEAD_CTX_seal() failed");
-        return NGX_ERROR;
-    }
-
-    EVP_AEAD_CTX_free(ctx);
-#else
-    int             len;
-    EVP_CIPHER_CTX  *ctx;
-
-    ctx = EVP_CIPHER_CTX_new();
-    if (ctx == NULL) {
-        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_CIPHER_CTX_new() failed");
-        return NGX_ERROR;
-    }
-
-    if (EVP_EncryptInit_ex(ctx, cipher, NULL, NULL, NULL) != 1) {
```

```
-       EVP_CIPHER_CTX_free(ctx);
-       ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_EncryptInit_ex() failed");
+       ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_CipherInit_ex() failed");
        return NGX_ERROR;
    }

@@ -509,28 +410,121 @@
        return NGX_ERROR;
    }

-   if (EVP_EncryptInit_ex(ctx, NULL, NULL, s->key.data, nonce) != 1) {
+   if (EVP_CipherInit_ex(ctx, NULL, NULL, s->key.data, NULL, enc) != 1) {
        EVP_CIPHER_CTX_free(ctx);
+       ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_CipherInit_ex() failed");
+       return NGX_ERROR;
+   }
+#endif
+
+   s->ctx = ctx;
+   return NGX_OK;
+}
+
+
+static ngx_int_t
+ngx_quic_crypto_open(ngx_quic_secret_t *s, ngx_str_t *out, u_char *nonce,
+    ngx_str_t *in, ngx_str_t *ad, ngx_log_t *log)
+{
+    ngx_quic_crypto_ctx_t  *ctx;
+
+    ctx = s->ctx;
+
+#ifdef OPENSSL_IS_BORINGSSL
+    if (EVP_AEAD_CTX_open(ctx, out->data, &out->len, out->len, nonce, s->iv.len,
+                          in->data, in->len, ad->data, ad->len)
+        != 1)
+    {
+        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_AEAD_CTX_open() failed");
+        return NGX_ERROR;
+    }
+#else
+    int  len;
+
+    if (EVP_DecryptInit_ex(ctx, NULL, NULL, NULL, nonce) != 1) {
+        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_DecryptInit_ex() failed");
+        return NGX_ERROR;
+    }
+
+    in->len -= NGX_QUIC_TAG_LEN;
+
+    if (EVP_CIPHER_CTX_ctrl(ctx, EVP_CTRL_AEAD_SET_TAG, NGX_QUIC_TAG_LEN,
+                            in->data + in->len)
+        == 0)
+    {
+        ngx_ssl_error(NGX_LOG_INFO, log, 0,
+                      "EVP_CIPHER_CTX_ctrl(EVP_CTRL_AEAD_SET_TAG) failed");
```

```
+        return NGX_ERROR;
+    }
+
+    if (EVP_CIPHER_mode(EVP_CIPHER_CTX_cipher(ctx)) == EVP_CIPH_CCM_MODE
+        && EVP_DecryptUpdate(ctx, NULL, &len, NULL, in->len) != 1)
+    {
+        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_DecryptUpdate() failed");
+        return NGX_ERROR;
+    }
+
+    if (EVP_DecryptUpdate(ctx, NULL, &len, ad->data, ad->len) != 1) {
+        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_DecryptUpdate() failed");
+        return NGX_ERROR;
+    }
+
+    if (EVP_DecryptUpdate(ctx, out->data, &len, in->data, in->len) != 1) {
+        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_DecryptUpdate() failed");
+        return NGX_ERROR;
+    }
+
+    out->len = len;
+
+    if (EVP_DecryptFinal_ex(ctx, out->data + out->len, &len) <= 0) {
+        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_DecryptFinal_ex failed");
+        return NGX_ERROR;
+    }
+
+    out->len += len;
+#endif
+
+    return NGX_OK;
+}
+
+
+ngx_int_t
+ngx_quic_crypto_seal(ngx_quic_secret_t *s, ngx_str_t *out, u_char *nonce,
+    ngx_str_t *in, ngx_str_t *ad, ngx_log_t *log)
+{
+    ngx_quic_crypto_ctx_t  *ctx;
+
+    ctx = s->ctx;
+
+#ifdef OPENSSL_IS_BORINGSSL
+    if (EVP_AEAD_CTX_seal(ctx, out->data, &out->len, out->len, nonce, s->iv.len,
+                          in->data, in->len, ad->data, ad->len)
+        != 1)
+    {
+        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_AEAD_CTX_seal() failed");
+        return NGX_ERROR;
+    }
+#else
+    int  len;
+
+    if (EVP_EncryptInit_ex(ctx, NULL, NULL, NULL, nonce) != 1) {
        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_EncryptInit_ex() failed");
```

```
        return NGX_ERROR;
    }

-   if (EVP_CIPHER_mode(cipher) == EVP_CIPH_CCM_MODE
+    if (EVP_CIPHER_mode(EVP_CIPHER_CTX_cipher(ctx)) == EVP_CIPH_CCM_MODE
        && EVP_EncryptUpdate(ctx, NULL, &len, NULL, in->len) != 1)
    {
-       EVP_CIPHER_CTX_free(ctx);
        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_EncryptUpdate() failed");
        return NGX_ERROR;
    }

    if (EVP_EncryptUpdate(ctx, NULL, &len, ad->data, ad->len) != 1) {
-       EVP_CIPHER_CTX_free(ctx);
        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_EncryptUpdate() failed");
        return NGX_ERROR;
    }

    if (EVP_EncryptUpdate(ctx, out->data, &len, in->data, in->len) != 1) {
-       EVP_CIPHER_CTX_free(ctx);
        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_EncryptUpdate() failed");
        return NGX_ERROR;
    }
@@ -538,7 +532,6 @@
    out->len = len;

    if (EVP_EncryptFinal_ex(ctx, out->data + out->len, &len) <= 0) {
-       EVP_CIPHER_CTX_free(ctx);
        ngx_ssl_error(NGX_LOG_INFO, log, 0, "EVP_EncryptFinal_ex failed");
        return NGX_ERROR;
    }
@@ -549,21 +542,32 @@
                        out->data + out->len)
        == 0)
    {
-       EVP_CIPHER_CTX_free(ctx);
        ngx_ssl_error(NGX_LOG_INFO, log, 0,
                      "EVP_CIPHER_CTX_ctrl(EVP_CTRL_AEAD_GET_TAG) failed");
        return NGX_ERROR;
    }

    out->len += NGX_QUIC_TAG_LEN;
-
-   EVP_CIPHER_CTX_free(ctx);
 #endif

    return NGX_OK;
 }


+void
+ngx_quic_crypto_cleanup(ngx_quic_secret_t *s)
+{
+    if (s->ctx) {
+#ifdef OPENSSL_IS_BORINGSSL
```

```
+       EVP_AEAD_CTX_free(s->ctx);
+#else
+       EVP_CIPHER_CTX_free(s->ctx);
+#endif
+       s->ctx = NULL;
+   }
+}
+
+
 static ngx_int_t
 ngx_quic_crypto_hp(ngx_log_t *log, const EVP_CIPHER *cipher,
     ngx_quic_secret_t *s, u_char *out, u_char *in)
@@ -666,6 +670,12 @@
         }
     }

+    if (ngx_quic_crypto_init(ciphers.c, peer_secret, is_write, log)
+        == NGX_ERROR)
+    {
+        return NGX_ERROR;
+    }
+
     return NGX_OK;
 }

@@ -675,10 +685,10 @@
     enum ssl_encryption_level_t level, ngx_uint_t is_write)
 {
     if (is_write == 0) {
-        return keys->secrets[level].client.key.len != 0;
+        return keys->secrets[level].client.ctx != NULL;
     }

-    return keys->secrets[level].server.key.len != 0;
+    return keys->secrets[level].server.ctx != NULL;
 }


@@ -686,8 +696,13 @@
 ngx_quic_keys_discard(ngx_quic_keys_t *keys,
     enum ssl_encryption_level_t level)
 {
-    keys->secrets[level].client.key.len = 0;
-    keys->secrets[level].server.key.len = 0;
+    ngx_quic_secret_t  *client, *server;
+
+    client = &keys->secrets[level].client;
+    server = &keys->secrets[level].server;
+
+    ngx_quic_crypto_cleanup(client);
+    ngx_quic_crypto_cleanup(server);
 }


@@ -699,6 +714,9 @@
```

```
    current = &keys->secrets[ssl_encryption_application];
    next = &keys->next_key;

+    ngx_quic_crypto_cleanup(&current->client);
+    ngx_quic_crypto_cleanup(&current->server);
+
    tmp = *current;
    *current = *next;
    *next = tmp;
@@ -762,6 +780,16 @@
        }
    }

+    if (ngx_quic_crypto_init(ciphers.c, &next->client, 0, c->log) == NGX_ERROR)
+    {
+        goto failed;
+    }
+
+    if (ngx_quic_crypto_init(ciphers.c, &next->server, 1, c->log) == NGX_ERROR)
+    {
+        goto failed;
+    }
+
    return;

 failed:
@@ -770,6 +798,23 @@
 }


+void
+ngx_quic_keys_cleanup(ngx_quic_keys_t *keys)
+{
+    ngx_uint_t          i;
+    ngx_quic_secrets_t  *next;
+
+    for (i = 0; i < NGX_QUIC_ENCRYPTION_LAST; i++) {
+        ngx_quic_keys_discard(keys, i);
+    }
+
+    next = &keys->next_key;
+
+    ngx_quic_crypto_cleanup(&next->client);
+    ngx_quic_crypto_cleanup(&next->server);
+}
+
+
 static ngx_int_t
 ngx_quic_create_packet(ngx_quic_header_t *pkt, ngx_str_t *res)
 {
@@ -801,8 +846,7 @@
    ngx_memcpy(nonce, secret->iv.data, secret->iv.len);
    ngx_quic_compute_nonce(nonce, sizeof(nonce), pkt->number);

-    if (ngx_quic_crypto_seal(ciphers.c, secret, &out,
```

```
-                         nonce, &pkt->payload, &ad, pkt->log)
+    if (ngx_quic_crypto_seal(secret, &out, nonce, &pkt->payload, &ad, pkt->log)
        != NGX_OK)
     {
        return NGX_ERROR;
@@ -862,13 +906,19 @@
     ngx_memcpy(secret.key.data, key, sizeof(key));
     secret.iv.len = NGX_QUIC_IV_LEN;

-    if (ngx_quic_crypto_seal(ciphers.c, &secret, &itag, nonce, &in, &ad,
-                         pkt->log)
+    if (ngx_quic_crypto_init(ciphers.c, &secret, 1, pkt->log) == NGX_ERROR) {
+        return NGX_ERROR;
+    }
+
+    if (ngx_quic_crypto_seal(&secret, &itag, nonce, &in, &ad, pkt->log)
        != NGX_OK)
     {
+        ngx_quic_crypto_cleanup(&secret);
        return NGX_ERROR;
     }

+    ngx_quic_crypto_cleanup(&secret);
+
     res->len = itag.data + itag.len - start;
     res->data = start;

@@ -999,7 +1049,7 @@
     u_char              *p, *sample;
     size_t              len;
     uint64_t            pn, lpn;
-    ngx_int_t           pnl, rc;
+    ngx_int_t           pnl;
     ngx_str_t           in, ad;
     ngx_uint_t          key_phase;
     ngx_quic_secret_t   *secret;
@@ -1088,9 +1138,9 @@
     pkt->payload.len = in.len - NGX_QUIC_TAG_LEN;
     pkt->payload.data = pkt->plaintext + ad.len;

-    rc = ngx_quic_crypto_open(ciphers.c, secret, &pkt->payload,
-                         nonce, &in, &ad, pkt->log);
-    if (rc != NGX_OK) {
+    if (ngx_quic_crypto_open(secret, &pkt->payload, nonce, &in, &ad, pkt->log)
+        != NGX_OK)
+    {
        return NGX_DECLINED;
     }

@@ -26,8 +26,10 @@

 #ifdef OPENSSL_IS_BORINGSSL
 #define ngx_quic_cipher_t           EVP_AEAD
+#define ngx_quic_crypto_ctx_t       EVP_AEAD_CTX
 #else
```

```
 #define ngx_quic_cipher_t           EVP_CIPHER
+#define ngx_quic_crypto_ctx_t       EVP_CIPHER_CTX
 #endif


@@ -48,6 +50,7 @@
     ngx_quic_md_t             key;
     ngx_quic_iv_t             iv;
     ngx_quic_md_t             hp;
+    ngx_quic_crypto_ctx_t    *ctx;
 } ngx_quic_secret_t;

@@ -100,14 +103,17 @@
     enum ssl_encryption_level_t level);
 void ngx_quic_keys_switch(ngx_connection_t *c, ngx_quic_keys_t *keys);
 void ngx_quic_keys_update(ngx_event_t *ev);
+void ngx_quic_keys_cleanup(ngx_quic_keys_t *keys);
 ngx_int_t ngx_quic_encrypt(ngx_quic_header_t *pkt, ngx_str_t *res);
 ngx_int_t ngx_quic_decrypt(ngx_quic_header_t *pkt, uint64_t *largest_pn);
 void ngx_quic_compute_nonce(u_char *nonce, size_t len, uint64_t pn);
 ngx_int_t ngx_quic_ciphers(ngx_uint_t id, ngx_quic_ciphers_t *ciphers,
     enum ssl_encryption_level_t level);
-ngx_int_t ngx_quic_crypto_seal(const ngx_quic_cipher_t *cipher,
-    ngx_quic_secret_t *s, ngx_str_t *out, u_char *nonce, ngx_str_t *in,
-    ngx_str_t *ad, ngx_log_t *log);
+ngx_int_t ngx_quic_crypto_init(const ngx_quic_cipher_t *cipher,
+    ngx_quic_secret_t *s, ngx_int_t enc, ngx_log_t *log);
+ngx_int_t ngx_quic_crypto_seal(ngx_quic_secret_t *s, ngx_str_t *out,
+    u_char *nonce, ngx_str_t *in, ngx_str_t *ad, ngx_log_t *log);
+void ngx_quic_crypto_cleanup(ngx_quic_secret_t *s);
 ngx_int_t ngx_quic_hkdf_expand(ngx_quic_hkdf_t *hkdf, const EVP_MD *digest,
     ngx_log_t *log);
```