МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Волинський національний університет імені Лесі Українки
Математичний факультет
Кафедра прикладної математики

«Теоретичні основи комп'ютерної безпеки та її організаційне забезпечення»

Методичні вказівки до лабораторних робіт

освітньо-професійної програми галузі знань
0403 «системні науки та кібернетика»

Луцьк-2010

УДК
ББК

Теоретичні основи комп'ютерної безпеки та її організаційне забезпечення. Методичні вказівки до лабораторних робіт. –, Луцьк 2010. – 49 с.

Розробник: старший викладач кафедри прикладної математики Волинського національного університету імені Лесі Українки Гопанчук С.О.

Рецензент: кандидат фізико-математичних наук, доцент кафедри фізики твердого тіла, С.А. Федосов

Рекомендовано до друку методичною комісією математичного факультету Волинського національного університету імені Лесі Українки
Протокол №1 від 30 серпня 2010 р.

Рекомендовано до друку методичною радою
Волинського національного університету імені Лесі Українки
Протокол № 1 від 22 вересня 2010р.

В методичних вказівках запропоновано лабораторні роботи для закріплення лекційного матеріалу із теоретичних основ комп'ютерної безпеки. Тематикою лабораторних робіт є моделі політик інформаційної безпеки. Структура лабораторної роботи містить короткі теоретичні відомості, постановку задачі та вимоги до захисту та оформлення роботи. Методичні рекомендації призначені для студентів галузі знань 0403 «системні науки та кібернетика».

In the proposed methodical pointing for laboratories working to consolidate the lecture material with the theoretical foundations of computer security. Subjects laboratory work is a model of information security policy. The structure of the laboratory work contains brief theoretical information, setting tasks and requirements for the protection and decoration work. Methodical recommendations are intended for students of the field of knowledge 0403 "system science and cybernetics.

ЗМІСТ

Вступ

Метою курсу по теоретичним основам комп’ютерної безпеки, призначеного для студентів спеціалізації «Технологія безпеки інформації», **є знайомство з поняттями захисту інформації, вивчення формальних моделей безпеки, політик безпеки, критеріїв і класів захищеності засобів обчислювальної техніки й автоматизованих інформаційних систем, розгляд стандартів по оцінці захищених систем, методів верифікації й методологія обстеження й проектування систем захисту.**

Слід зазначити, що зміст курсу відображає дійсний стан теорії комп'ютерної безпеки й, не претендуючи на її повне охоплення, представляє підхід до викладання положень теорії. Розглянуті в рамках курсу теоретичні основи не несуть у собі відповідей на питання, як захищати конкретну систему. Разом з тим, вони включають загальну методологію, що дозволяє визначити, якими захисними властивостями повинна володіти проектована система, яким вимогам вона повинна відповідати, яким чином повинна проводитися її розробка.

Метою **лабораторних робіт** з курсу теоретичних основ інформаційної безпеки є отримання студентами практичного досвіду по оцінці загроз та вразливостей, застосуванню моделей розмежування доступу, формуванню політик безпеки й по інших аспектах теоретичних основ безпеки. Саме на лабораторних заняттях студентам дається можливість зрозуміти важливість теоретичної сторони захисту інформації як основи, на якій базуються практичні рішення. Для кращої наочності та отримання ширших практичних навиків необхідно розробка та створення комплексних програмних продуктів, що реалізують теоретичні моделі.

Програмний додаток, що розробляється студентом повинно являти собою, віртуальне середовище у якому можливо віртуально задавати різні умови поводження й взаємодії суб'єктів і об'єктів, тобто реалізовані різні моделі розмежування доступу до інформації. Крім того, повинен передбачатися аналіз загроз і захищеності автоматизованої системи.

Таким чином, саме програмне середовище повинне являти собою емуляцію операційної системи на зразок віртуальної машини, у якій можна вибрати різні ситуації у відповідність із розглянутою темою - моделлю розмежування.

Щораз у програмній лабораторії студент вибирає певне середовище, що представляє ту або іншу модель.

Передбачуваний зміст лабораторних робіт.

1. Суб’єктно-об'єктні моделі. Це середовище для всього класу суб'єктних моделей розмежування доступу. Ціль роботи: закріплення теоретичного матеріалу по даній моделі політики безпеки, дослідження заданої системи на відповідність вимогам заданої політики безпеки. Користувачеві видається множина елементів, з яких складається системи. Пропонується розділити їх на об'єкти й суб'єкти по ознаці активності; виділити в системі спеціальні суб'єкти - Монітор безпеки об'єктів (МБО), Монітор безпеки суб’єктів (МБС); сформувати матрицю доступу, задавши права доступу; визначити домени захисту; перевірити коректність суб'єктів відносно один одного, МБС і МБО; виділити ті суб’єкти, які можуть утворити Ізольоване програмне середовище (ІПС).

2. Дискреційна модель. Дане середовище - реалізація дискреційна модель безпеки. Студентам пропонується для заданої матриці доступів суб'єктів до об'єктів мінімізувати привілею суб'єктів, доповнити матрицю доступів тимчасовими доменами у відповідність із завданням. Також дане середовище повинна містити окрему роботу з порушення дискреційної політики безпеки за допомогою програми типу «троянський кінь». Особливий аналіз класу дискреційних моделей проводиться на основі моделі Take-Grant. Студентові видається матриця доступів і пропонується за допомогою спеціальних візуальних компонентів середовища графічно задати дану ситуацію у вигляді графа переходів. Друга частина складається в перевірці можливості доступу від одного суб'єкта до іншого набору декількох графів переходів (для частини типових завдань - з використанням теореми про tg-зв'язний шлях, а для частини простих завдань - без її використання). Програма повинна мати можливість перевірки

вірності побудови графа переходів і відповідей студента по можливості доступу до заданого об'єкта.

3. Мандатна модель. Як приклад реалізації мандатної моделі інформаційної безпеки розглядається модель Белла-Лападула. Ціль роботи: ознайомлення із класичною мандатною моделлю доступу для захисту від загроз розкриття інформації; реалізація загрози розкриття інформації заснованої на декласифікації об'єкта від секретного до несекретного (також пропонується сформулювати методи протидії даній атаці); завдання рівнів таємності, зниження рівня таємності об'єкта.

Хід роботи: пропонується задати в середовищі декілька суб'єктів, об'єктів і рівнів секретності. Для кожного користувача скласти список документів, доступних йому для роботи за умови, що користувач не знижує свого рівня допуску. Перевірити можливість доступу від імені різних створених суб'єктів до об'єктів, що мають рівень таємності менший, більший або дорівнює рівню таємності суб'єкта. Показати на прикладі, що мандатна політика не може бути порушена програмою типу «троянський кінь».

4. Рольове керування доступом. Ціль роботи: дослідження можливостей, що представляються моделями безпеки з рольовим керуванням доступом.

З основною концепцією ролі пов’язано два аспекти: ролям призначаються користувачі й ролям призначаються привілеї й дозволи. Користувач, якому призначено роль, таким чином, отримує привілеї й дозволи цієї ролі. Далі необхідно переконатися в захисті інформації в системі з реалізованим рольовим керуванням доступом: 1) виконати для декількох пар суб’єкт-об'єкт ті операції, які визначені у файлі опису моделі і які мають правильну умову доступу: Переконатися в успішній реалізації; 2) спробувати виконати інші операції: переконатися, що модель захищає від подібних несанкціонованих дій.

Крім даних розділів в програмному середовищі реалізуються моделі із наступних розділів інформаційної безпеки: аналіз загроз для заданої системи; дискреційна модель HRU; модель доменів і ін.

Лабораторна робота № 1.

Тема роботи: Реалізація дискреційної механізму управління доступом.

Мета роботи: програмно реалізувати одну із моделей дискреційного управління доступом, проаналізувати недоліки та переваги, що притаманні цим моделям, показати вразливість дискреційних моделей відносно програми типу «троянський кінь».

Короткі теоретичні відомості.

Основою **дискреційної політики безпеки** (ДПБ) є дискреційне управління доступом (Discretionary Access Control – DAC), яке визначається двома властивостями:

- всі суб'єкти і об'єкти повинні бути однозначно ідентифіковані;
- права доступу суб'єкта до об'єкта системи визначаються на основі деякого зовнішнього відносно системи правила.

ДПБ реалізується за допомогою матриці доступу (access matrix), яка фіксує множину кожного суб'єкта до доступних йому об'єктів та суб'єктів.

Існує декілька варіантів задання матриці доступу.

1. Листи можливостей (privilege list, profile): для кожного суб'єкта створюється лист (файл) усіх об'єктів, до якого він має доступ.

2. Листи контролю доступу (access control list): для кожного об'єкта створюється список усіх суб'єктів, що мають доступи до нього.

Моделі дискреційного управління доступом.

Модель розповсюдження прав доступу Take – Grant.

Модель Take – Grant використовується для аналізу систем захисту, що реалізують дискретну ПБ. Ціль моделі – дати відповідь на запитання про можливість отримання прав доступу суб'єктом системи на об'єкт в стані, що описується графом доступу.

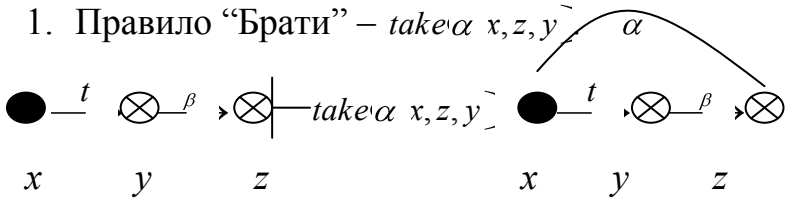
Опишемо формально дану модель. Позначимо: ***O*** – множина об'єктів системи (наприклад файли чи об'єкти пам'яті); ***S*** – множина суб'єктів системи (користувачі, процеси)($S \subseteq O$); ***R***= $\{r_1, r_2, ..., r_m\} \cup \{t, g\}$ – множина прав доступу суб'єктів до об'єктів, де *t(take)*– право брати права доступу, *g(grant)* – право

давати права доступу; *G=(S,O,E)* – скінчений, помічений орієнтований граф, що показує поточні доступи у системі. Елементи множини *S* позначаються •, а множини *O* – ⊗; $E \subseteq$)×)× ? – позначається дугами графа.

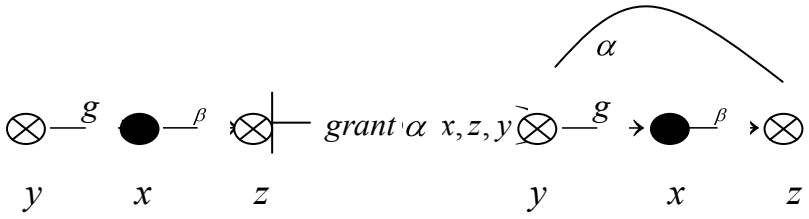
Стан системи описується графом доступів. Перехід системи із стану в стан визначається операціями чи правилами перетворення графа доступу.

В класичній моделі Take – Grant чотири правила перетворення графа. Нехай $x \in S, y, z \in O, \quad \beta \subseteq R, \alpha \subseteq$?.

1. Правило "Брати" – *take*(α x,z,y



2.Правило "Давати" – *grant*(α :,y,z).



3.Правило " Створити" – *greate*(β x,y). $y \in$) – новий об'єкт чи суб'єкт.

4.Правило "Вилучити" – *remove*(α :,y). Суб'єкт *x* вилучає право доступу α на об'єкт *y*.

В даній моделі визначаються умови санкціонованого та несанкціонованого отримання прав доступу.

Нехай $x, y \in$)- різні об'єкти графа доступу $G_0 = (S_0, O_0, E_0), \alpha \in$?. Предикат "можливий доступ" $(\alpha :, y, z)$ істинний тоді і лише тоді коли існують графи $G_1 = (S_1, O_1, E_1), ..., G_N = (S_N, O_N, E_N)$, що перетворюються один в одного та $(x, y, \alpha \in {}_N$.

Означення 1.1. Вершини графа доступів є *tg* – зв'язними, якщо в графі між ними існує такий шлях, що кожна дуга цього шляху помічена *t* або *g*.

Означення 1.2. Островом в довільному графі G_0 називається його максимальний *tg*-зв'язний підграф, що складається лише з вершин суб'єктів.

Теорема 1.1. Нехай $G_0 = (S_0, O_0, E_0), \alpha \in ?$ – довільний граф доступів. Предикат "можливий доступ" $(\alpha\ x, y, G_0)$ істинний тоді і лише тоді коли виконуються умови:

- існують об'єкти $s_1, ..., s_m$, такі що кожен з них має доступ $\gamma_i, i = \overline{,m}$ до об'єкта y і $\alpha = \bigcup_{i=}^{m} \lambda$;
- існують суб'єкти $x_1', ..., x_m', s_1', ..., s_m'$ такі що:

- $x = s_i'$, або x_i' з'єднаний з x через об'єкти, що пов'язані між собою tg – зв'язками з виглядом $\overrightarrow{t}^* \overrightarrow{g}$, де $*$ – повторення;

- $s_i = s_i'$, або s_i' з'єднане з s_i через об'єкти, що пов'язані між собою tg – зв'язками з виглядом $\overrightarrow{t}^*$;

- кожна пара (x_i', s_i') з'єднана островами.

В моделі також показано можливість викрадення доступу [1], [5], [8].

Модель матриці доступів HRU.

Модель HRU використовується для аналізу систем захисту, що реалізують дискретну ПБ. При цьому система захисту представлена скінченим автоматом, що функціонує відповідно з певними правилами переходу.

Позначимо: ***O*** – множина об'єктів системи; ***S*** – множина суб'єктів системи ($S \subseteq O$); ***R*** – множина прав доступу суб'єктів до об'єктів, зокрема право на читання (read), право на запис (write) та право на володіння (own); ***M*** – матриця доступів, рядки якої відповідають суб'єктам, а стовпці об'єктам; $M[s, o] \subseteq ?$ – права доступу суб'єкта s до об'єкта o.

Окремий автомат, побудований згідно моделі HRU буде називатись системою. Функціонування системи розглядається лише з точки зору зміни в матриці доступів. Можливі зміни визначаються шістьма примітивними операторами:

- "Внести" право $r \in ?$ в $M[s, o]$ – отримання суб'єктом s права доступу r на об'єкт o.

- "Вилучити" право $r \in \wr$ з M $\blacksquare, o$ – вилучення у суб'єкта s права r на об'єкт o.
- "Створити" суб'єкт s' – додавання в систему нового суб'єкта s' . При цьому в матрицю доступу додаються новий рядок та стовпець.
- "Створити" об'єкт o' – додавання в систему нового об'єкта o'. При цьому в матрицю доступу додається новий стовпець.
- "Знищити" суб'єкт s' – вилучення із системи суб'єкта s'. При цьому з матриці доступу вилучається відповідний рядок та стовпець.
- "Знищити" об'єкт o' – вилучення із системи об'єкта o' . При цьому з матриці доступу вилучається відповідний стовпець.

В результаті виконання примітивного оператора α здійснюється перехід системи із стану $Q=(S,O,M)$ в новий стан $Q' = (S', O', M')$.

Із примітивних операторів може складатись команда. Кожна команда складається з двох частин:

- умова, при якій виконується команда;
- послідовність примітивних операторів.

Означення 1.3 Будемо рахувати, що можливе витікання права $r \in \wr$ в результаті виконання команди *c,* якщо при переході системи в стан Q' виконується примітивний оператор, що вносить r в елемент матриці доступів M, який до цього в ній не містився.

Означення 1.4. Початковий стан Q_0 називається безпечним по відношенню до деякого права $r \in \wr$, якщо неможливий перехід системи в такий стан Q в якому може виникнути витікання права $r \in \wr$.

Означення 1.5 Система називається монооперaційною, якщо кожна команда виконує один примітивний оператор.

Теорема 1.2 Існує алгоритм, що перевіряє чи є початковий стан монооперaційної системи безпечний для даного права $r \in \wr$.

Теорема 1.3 Задача перевірки безпеки будь – яких систем алгоритмічно не вирішується.

Перелік знань та умінь, яким повинен оволодіти студент.

У процесі виконання лабораторної роботи студент повинен знати:

- призначення та структура дискреційного управління доступом;
- моделювання дискреційного управління доступом;
- вразливості дискреційних моделей;
- складності у реалізації дискреційних моделей та сфери їх застосування.

Після виконання лабораторної роботи студент повинен вміти:

- будувати алгоритми для програмної реалізації дискреційних моделей управління доступом;
- складати технічне завдання на програмну розробку;
- програмно реалізовувати моделі HRU та Take-Grant;
- реалізовувати канали витоку інформації в системах захисту, що реалізують дані моделі;
- формувати звіти.

Вимоги до програми лабораторної роботи.

Порядок виконання роботи

Постановка завдання: створити операційну оболонку, в якій реалізовано дискреційне управління доступом та реалізуються наступні функції:

- реєстрація суб'єктів;
- авторизація суб'єктів;
- здійснення операцій суб'єкта над об'єктами;
- реалізація ядра безпеки на основі однієї із моделей дискреційного розмежування доступу;
- контроль доступів суб'єктів до об'єкті в режимі «реальному» часі з допомогою ядра безпеки;
- вести журнал аудиту подій.

Етапи виконання:

1. Ознайомитись із теоретичними положеннями моделей дискреційного управління доступом;
2. Вибрати середовище програмування та обґрунтувати вибір;

3. Створити модуль реєстрації та авторизації суб’єктів згідно вибраної моделі;

4. Розробити модуль однозначної ідентифікації об’єктів;

5. Розробити процедури, що дозволяють здійснювати операції суб’єктам над об’єктами;

6. Створити модуль «адміністратора» в якому здійснюється управління суб’єктами, об’єктами та правами доступу згідно вибраної моделі;

7. Реалізувати модуль перевірки права на доступ, який хоче отримати суб’єкт до об’єкта в режимі «реальному» часу згідно вибраної моделі (ядро безпеки);

8. Реалізувати модуль аудиту подій, що дозволяє протоколювати на вибір: всі події, неуспішні доступи, успішні операції.

Бажаний результат роботи

Результатом виконання лабораторної роботи є:

- закінчений програмний продукт (операційна оболонка), що реалізує всі зазначенні у постановці завдання функції;
- звіт, що подається у електронному та друкованому вигляді (порядок оформлення звіті нижче по тексту);
- лістинг програмного продукту із коментарями, що роздруковується після успішного запуску програми на виконання.

Форма звітності.

Звіт повинен подаватись в електронному та друкованому вигляді і повинен містити наступні розділи:

- короткий опис проблем програмної реалізації вибраної моделі;
- обґрунтування вибору середовища програмування;
- специфікація програмного продукту:

- призначення розробки;

- вимоги до програмного продукту:

- *вимоги до функціональних характеристик.*

- *умови експлуатації.*

- *вимоги до інформаційної та програмної сумісності.*

- техніко-економічні показники програми;

- керівництво користувача;

- тестування програми;

- висновок.

Варіанти завдань.

При реалізації операційної оболонки за основу слід вибрати одну із запропонованих моделей дискреційного управління доступом згідно отриманих варіантів.

Варіант 1. Автоматна модель HRU.

Варіант 2. Модель розповсюдження прав Take-Grant.

Варіант 3. Розширена модель Take-Grant.

Варіант 4. Об'єкто-орієнтована модель ізольованого-програмного середовища.

Варіант 5. Модель дискреційного ядра безпеки.

Список джерел

1. Щеглов Ю.А. Защита компьютерной информации от несанкционированного доступа. – СПб: Наука и Техника, 2004. – 384 с.

2. Теоретические основы компютерной безопасности. / П.Н. Девянин, О.О. Михальский, Д.И Правиков и др. – М.: Радио и связь, 2000. – 192 с.

3. Девянин П.Н Модели безопасности компьютерных систем: Учебн. пособие для студентов высш. учеб. завед. – М.: Издательский центр «Академия», 2005. – 144 с.

4. Куприянов А.И. Основы защиты информации.- М.: Издательский центр «Академия», 2006. – 256 с.

5. Малюк А.А. Информационная безопасность:концептуальные и методологические основы защиты информации. Учеб. Пособие для вузов.- М.: Горячая линия. – Телеком, 2004 .- 208 с.

Лабораторна робота № 2.

Тема роботи: Реалізація мандатного механізму управління доступом.

Мета роботи: програмно реалізувати одну із моделей мандатного управління доступом, проаналізувати недоліки та переваги, що притаманні цим моделям, показати стійкість мандатних систем відносно шкідливих програм.

Короткі теоретичні відомості.

Основу **мандатної (повноважної) політики безпеки** (МПБ) складає мандатне управління доступом (Mandatory Access Control – MAC), яке має на увазі, що:

- всі суб'єкти і об'єкти повинні бути однозначно ідентифіковані;
- заданий лінійно упорядкований набір міток секретності;
- кожному об'єкту системи привласнена мітка секретності, яка визначає цінність інформації, що міститься в ньому – його рівень секретності в АС;
- кожному суб'єкту системи привласнена мітка секретності, яка визначає рівень довіри до нього в АС – максимальне значення мітки секретності об'єктів, до яких суб'єкт має доступ; мітка секретності суб'єкта називається його рівнем доступу.

Основна ціль МПБ – запобігання витоку інформації від об'єктів з високим рівнем доступу до об'єктів з низьким рівнем доступу, тобто протидія виникненню в АС інформаційних каналів зверху вниз. Вона оперує, таким чином, поняттями інформаційного потоку і цінності (певним значенням мітки секретності) інформаційних об'єктів.

Найчастіше МПБ описують в термінах, поняттях і визначеннях властивостей моделі Белла-Лападула. В рамках цієї моделі доводиться важливе твердження, яке вказує на принципіальну відміну систем, що реалізують мандатний захист, від систем з дискреційним захистом: **якщо початковий стан системи безпечний, і всі переходи системи з стану в стан не порушують обмежень, сформульованих ПБ, то будь-який стан системи безпечний.**

Однак МПБ має дуже серьйозні вади – вона складна для практичної реалізації і вимагає значних ресурсів обчислювальної системи. Це пов'язано з

тим, що інформацйних потоків в системі величезна кількість і їх не завжди можна ідентифікувати. Саме ці вади часто заважають її практичному використанню.

Модель системи безпеки Белла-Лападула.

Класична модель Белла-Лападула (БЛ) побудована для аналізу систем захисту, що аналізують мандатне розмежування доступу.

Позначимо: ***O*** – множина об'єктів системи (наприклад всі системні файли); ***S*** – множина суб'єктів системи (наприклад користувачі та процеси); *R={read, write,append, execute}* – множина видів доступу суб'єктів з *S* до об'єктів з *O*; $B = b \subseteq ' \times) \times ?\}$ – множина можливих множин поточних доступів; ***M*** – матриця доступів, рядки якої відповідають суб'єктам, а стовпці об'єктам; $M \blacksquare, o \bar{} \subseteq$! – права доступу суб'єкта *s* до об'єкта *o; L* – множина рівнів секретності; $f_s : S \rightarrow$ – рівень допуска суб'єкта; $f_o : O \rightarrow$ – рівень секретності об'єкта; $f_c : S \rightarrow$ – поточний рівень допуску суб'єкта при цьому $\forall \in f_s \blacksquare \bar{} \leq f_c \blacksquare \bar{}$; *H* – поточний рівень ієрархії об'єктів; *V* – множина станів системи; *Q* – множина запитів системи*;* *D* – множина рішень по запитам; $W \subseteq) \times) \times '^2$ – множина дій системи, де $\blacksquare, d, v_1, v_2 \bar{} \in '$ означає що система по запиту *q* з відповідю *d* перейшла із стану v_1 в стан v_2; *X* – множина функцій що задають всі можливі запитів до системи; *Y* – множина функцій що задають всі можливі відповіді по запитам; *Z* – множина функцій що задають всі можливі послідовності станів системи.

За допомогою означень та теорем опишемо як визначається безпека у системі та за допомогою чого вона досягається.

Означення 2.1. $\Sigma \ ), D, W, z_0 \bar{} \subseteq \ ' \times ' \times ?$ називається системою якщо $\blacksquare, y, z \bar{} \in \ ), D, W, z_0 \bar{}$ тоді і лише тоді коли $\blacksquare_t y_t z_{t+1} z_t \bar{} \in '$ для кожного *t*, z_0 – початковий стан системи.

Безпека системи визначається за допомогою трьох властивостей.

Означення 2.2. Доступ $\blacksquare, o, r \bar{} \in \times) \times ?$ володіє *ss*-властивістю відносно $f = \blacksquare_s f_o f_c \bar{}$, якщо виконуються одна з умов:

- $r \in \{execute, append\}$;
- $r \in \{read, write\}$ та $f_s \blacksquare \bar{} \leq f_o \blacksquare \bar{}$.

Означення 2.3. Доступ $(s,o,r) \in S \times O \times R$ володіє *-властивістю відносно $f = (f_s, f_o, f_c)$, якщо виконуються одна з умов:

- *r=execute;*
- *r=append* та $f_o(o) \geq f_c(s)$;
- *r=read* та $f_o(o) \leq f_c(s)$;
- *r=write* та $f_o(o) = f_c(s)$.

Означення 2.4. Стан системи $(b,M,f,h) \in V$ володіє *ds*-властивістю, якщо $\forall (s,o,r) \in B \Rightarrow (s,o,r) \in M_{so}$.

Далі вкажемо теореми за допомогою яких можна перевіряти та обгрунтовувати умови безпеки системи.

Теорема 2.1. Система $\Sigma(R,D,W,z_0)$ володіє *ss*-властивістю для будь-якого початкового стану z_0, що володіє *ss*-властивістю тоді і лише тоді, коли $\forall v, d(b^*, M^*, f^*, h^*), (b,M,f,h) \in W$ задовільняє умовам:

Умова 1. $\forall (s,o,r) \in b^* \setminus b$ володіє *ss*-властивістю відносно f^*.

Умова 2. Якщо $(s,o,r) \in$ не володіє *ss*-властивістю відносно f^*, то $(s,o,r) \notin b^*$.

Теорема 2.2. Система $\Sigma(R,D,W,z_0)$ володіє *-властивістю відносно $S' \subseteq S$ для будь-якого початкового стану z_0, що володіє *-властивістю тоді і лише тоді, коли $\forall v, d(b^*, M^*, f^*, h^*), (b,M,f,h) \in W$ задовільняє умовам:

Умова 1. $\forall s \in S'$, $\forall (s,o,r) \in b^* \setminus b$ володіє *-властивістю відносно f^*.

Умова 2. $\forall s \in S'$, якщо $(s,o,r) \in$ не володіє *-властивістю відносно f^*, то $(s,o,r) \notin b^*$.

Теорема 2.3. Система $\Sigma(R,D,W,z_0)$ володіє *ds*-властивістю для будь-якого початкового стану z_0, що володіє *ds*-властивістю тоді і лише тоді, коли $\forall v, d(b^*, M^*, f^*, h^*), (b,M,f,h) \in W$ задовільняє умовам:

Умова 1. $\forall (s,o,r) \in b^* \setminus b$ виконується $r \in M_{so}$.

Умова 2. Якщо $(s,o,r) \in$ та $r \notin M_{so}$, то $(s,o,r) \notin b^*$.

Теорема BST (Basic Securitu Theorem). Система $\Sigma(D,W,z_0)$ безпечна тоді та лише тоді коли z_0 безпечний стан і множина дій *W* задовільняє умовам теорем 2.1,2.2,2.3 [1], [9].

Модель безпеки інформаційних потоків.

Розглянемо систему захисту $\sum$, що реалізує мандатне розмежування доступу, та носить назву моделі безпеки інформаційних потоків. Без обмеження загальності можна рахувати, що :

- в системі використовуються два рівня: секретності: високий та низький;
- всі об'єкти поділяються на високорівневі *(H)*, та низькорівневі *(L)*;
- всі взаємодії між ними здійснюються через систему захисту $\sum$.

Таким чином схема системи буде виглядати так (рис.2.1):

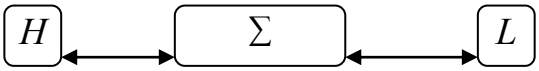


Рис.2.1 Схема системи захисту.

Задача системи захисту – не допустити виникнення інформаційного потоку від високорівневих об'єктів до низькорівневих.

Нехай *H* та *L* – випадкові величини.

Означення 2.5. В системі присутній інформаційний потік від високорівневих об'єктів *H* до низькорівневих *L*, якщо деяке можливе значення змінної в деякому стані низькорівневого об'єкту неможливе одночасно з можливим станом змінних станів високорівневих об'єктів, якщо такі потоки відсутні то система називається безпечною.

У формальному вигляді це означення у наступним чином: між *H* та *L* потік відсутній, якщо $p(H)>0, p(L)>0$, то $p(H/L)>0$. Так як при $p(H)>0, p(L)>0$ виконується $p(L/H)=p(H,L)/p(H)=p(H/L)p(L)/p(H)$, то в умовах означення 10 з істиності нерівності $p(H/L)>0$ слідує істиність $p(L/H)>0$, що вказує на відсутність інформаційних потоків від низькорівневих до високорівневих об'єктів.

Якщо ввести параметр часу то потік між об'єктами двох рівнів може виникати таким чином. Значення низькорівневих об'єктів в момент часу *t* може бути зафіксоване в високорівневому об'єкті в момент часу *t+1,* та навпаки.

З врахуванням того що стан системи впливає на наступний стан лише через інформацію, що зберігається в об'єктах системи дамо означення.

Означення 2.6 Система безпечна в змісті інформаційного невтручання, якщо виконується рівність

$p(L_t / H_s, L_s) = p(L_t / L_s)$, де *s,t=0,1,2, s<t* – моменти часу. [3]

Перелік знань та умінь, якими повинен оволодіти студент.

У процесі виконання лабораторної роботи студент повинен знати:

- призначення та структура мандатного управління доступом;
- проблеми моделювання мандатного управління доступом;
- положення, що доводять гарантованість мандатного управління безпекою;
- складності у реалізації мандатних моделей та сфери їх застосування;
- перелік моделей, що реалізують мандатну політику безпеки.

Після виконання лабораторної роботи студент повинен вміти:

- будувати алгоритми для програмної реалізації мандатних моделей управління доступом;
- складати технічне завдання на програмну розробку;
- програмно реалізовувати моделі, що реалізують мандатну політику безпеки;
- працювати з API-функціями операційної системи;
- реалізовувати алгоритми оцінки цінності об'єктів та суб'єктів;
- формувати звіти.

Вимоги до програми лабораторної роботи.

Порядок виконання роботи

Постановка завдання: створити операційну оболонку, в якій реалізовано мандатне управління доступом та реалізуються наступні функції:

- реєстрація та ідентифікація суб'єктів;
- надання рівня довіри суб'єкту;

- однозначна ідентифікація об’єкта;
- визначення рівня цінності об’єкта;
- аналіз кожного процесу, що породжується та заборона породження від суб’єктів з нижчим рівнем довіри до об’єктів з вищим рівнем цінності;
- ядро безпеки оболонки на основі однієї із моделей мандатного управління доступом;
- вести журнал аудиту подій.

Етапи виконання:

1. Ознайомитись із теоретичними положеннями моделей, що реалізують мандатну політику безпеки;
2. Вибрати середовище програмування та обґрунтувати вибір;
3. Створення модуля реєстрації та ідентифікації суб’єктів;
4. Створення модуля, що реалізує алгоритм надання рівня доступу суб’єкту;
5. Реалізація модуля однозначної ідентифікації об’єкта;
6. Розробка модуля визначення рівня цінності об’єкта;
7. Створення модуля аналізу процесів та заборони процесів від суб’єктів з нижчим рівнем довіри до об’єктів з вищим рівнем цінності;
8. Створення монітору безпеки, що регулює доступи суб’єктів до об’єктів згідно однієї із моделей мандатного управління доступом;
9. Створення модуля аудиту подій доступу із можливістю вибрати тип подій, які необхідно протоколювати.

Бажаний результат роботи

Результатом виконання лабораторної роботи є:

- закінчений програмний продукт (операційна оболонка), що реалізує всі зазначенні у постановці завдання функції;
- звіт, що подається у електронному та друкованому вигляді (порядок оформлення звіті нижче по тексту);
- лістинг програмного продукту із коментарями, що роздруковується після успішного запуску програми на виконання.

Форма звітності.

Звіт повинен подаватись в електронному та друкованому вигляді і повинен містити наступні розділи:

- короткий опис проблем програмної реалізації вибраної моделі;
- обґрунтування вибору середовища програмування;
- специфікація програмного продукту:

- призначення розробки;

- вимоги до програмного продукту:

- *вимоги до функціональних характеристик.*

- *умови експлуатації.*

- *вимоги до інформаційної та програмної сумісності.*

- техніко-економічні показники програми;

- керівництво користувача;

- тестування програми;

- висновок.

Варіанти завдань.

При реалізації операційної оболонки за основу слід вибрати одну із запропонованих моделей мандатного управління доступом згідно отриманих варіантів.

Варіант 1. Класична модель Белла-ЛаПадула.

Варіант 2. Модель мандатної політики цілісності інформації Біба.

Варіант 3.Модель системи військових повідомлень.

Варіант 4. Модель безпеки інформаційних потоків.

Варіант 5. Модель «китайської стіни».

Варіант 6. Модель CWM.

Список джерел

1. Щеглов Ю.А. Защита компьютерной информации от несанкционированного доступа. – СПб: Наука и Техника, 2004. – 384 с.
2. Теоретические основы компютерной безопасности. / П.Н. Девянин, О.О. Михальский, Д.И Правиков и др. – М.: Радио и связь, 2000. – 192 с.

3. Девянин П.Н Модели безопасности компьютерных систем: Учебн. пособие для студентов высш. учеб. завед. – М.: Издательский центр «Академия», 2005. – 144 с.
4. Куприянов А.И. Основы защиты информации.- М.: Издательский центр «Академия», 2006. – 256 с.
5. Малюк А.А. Информационная безопасность:концептуальные и методологические основы защиты информации. Учеб. Пособие для вузов.- М.: Горячая линия. – Телеком, 2004 .- 208 с.
6. Запечников С. В., Милославская Н. Г., Толстой А. И., Ушаков Д. В. И74 Информационная безопасность открытых систем: Учебник для вузов. В 2-х томах. Том 2 – Средства защиты в сетях / – М.: Горячая линия–Телеком, 2008. – 558 с.
7. Мещеряков Р.В., Праскурин Г.А. Теоретические основы компьютерной безопасности. Курс лекций. Раздел 1. Томск: Изд-во ТУСУР, 2005. – 147 с.
8. Мещеряков Р.В., Праскурин Г.А. Теоретические основы компьютерной безопасности. Курс лекций. Раздел 2. Томск: Изд-во ТУСУР, 2005. – 230 с.

Лабораторна робота № 3.

Тема роботи: Реалізація рольового механізму управління доступом.

Мета роботи: програмно реалізувати одну із моделей рольового управління доступом, проаналізувати проблеми реалізації відношень та функцій, що визначають суть рольових моделей, показати переваги над мандатними та дискреційними моделями.

Короткі теоретичні відомості.

Рольова політика безпеки (РПБ) (Role Base Access Control – RBAC) - здійснюється як на основі матриці прав доступу для ролей, так і за допомогою правил, які регламентують призначення ролей користувачам та їх активацію під час сеансів.

В РПБ класичне поняття **суб'єкт** заміщується поняттями **користувач** і **роль.** Користувач – це людина, яка працює з системою і виконує певні службові обов'язки. Роль – це активно діюча в системі абстрактна суттєвість, з якою пов'язаний обмежений, логічно зв'язаний набір повноважень, які необхідні для здійснення певної діяльності.

Дана ПБ виходить із твердження, що користувачі, що працюють в системі, діють не від свого особистого імені – вони завжди здійснюють певні службові обов'язки, тобто виконують деякі ролі, які аж ніяк не пов'язані з їх особистістю.

В такій ситуації РПБ дозволяє розподілити повноваження між цими ролями відповідно до їх службових обов'язків: ролі адміністратора призначаються спеціальні повноваження, які дозволять йому контролювати роботу системи і керувати її конфігурацією, роль менеджера баз даних дозволяє здійснювати керування сервером БД, а права простих користувачів обмежуються мінімумом, необхідним для запуску прикладних програм.

В моделі РПБ визначаються наступні множини:

- U - множина користувачів;
- R - множина ролей;
- P – множина повноважень на доступ до об'єктів, що представляється, наприклад, у вигляді матриці прав доступу;

- S – множина сеансів роботи користувачів з системою.

Для перелічених множин визначаються наступні відношення:

$PA \subseteq ? \times ?$ – відображає множину повноважень на множину ролей, встановлюючи для кожної ролі набір наданих їй повноважень;

$UA \subseteq / \times ?$ – відображає множину користувачів на множину ролей, визначаючи для кожного користувача набір доступних йому ролей.

Правила керування доступом рольової політики безпеки визначаються наступними функціями:

user: *S->U* – для кожного сеанса *s* ця функція визначає користувача *u*, який здійснює цей сеанс роботи з системою: *user(s)=u*;

roles: *S->R* – для кожного сеанса *s* ця функція визначає набір ролей з множини *R*, що можуть бути одночасно доступні користувачу *u* в цьому сеансі: *roles(s)={r | (user(s),r) ∈UA};*

permissions: *S->P* – для кожного сеанса *s* ця функція задає набір доступних в ньому повноважень, який визначається як сукупність повноважень всіх ролей, що приймають участь в цьому сеансі: *permissions(s)={p | (p,r) ∈ PA }*.

В якості критерію безпеки рольової моделі використовується наступне правило: **система вважається безпечною, якщо будь-який користувач системи, що працює в сеансі s, може здійснити дії, які вимагають повноважень p тільки в тому випадку, коли** $p \in permissions(s)$.

З формулювання критерію безпеки моделі РПБ виникає, що управління доступом здійснюється головним чином не за допомогою призначення повноважень ролям, а шляхом задання відношення *UA*, яке призначає ролі користувачам, і функції *roles*, що визначає доступний в сеансі набір ролей. Тому числені інтерпретації рольової моделі відрізняються видом функцій *user*, *roles* і *permission*, а також обмеженнями, що накладаються на відношення *PA* та *UA*.

Базова модель РРД.

В базовій моделі основні множини та відношення вважаються незмінними.

Множина ролей на які авторизуються користувачі під час одного сеансу модифікується самим користувачем. В базовій моделі РРД одна сесія не може породити іншу сесію, її може породити лише користувач.

Означення. Ієрархією ролей в базовій моделі РРД називають задане на множині ролей відношення часткового порядку $\leq$ (рифлексифність, антисиметрічність, транзитивність) і при цьому виконується умова

$$u \in \ ,(r,r^{\backslash} \in \ ,r \in \ A(u) \wedge \ ^{\backslash} \leq \ ) \Rightarrow \ ^{\backslash} \in \ A(u))$$

Це означає, що коли користувач авторизований до певної ролі, то він автоматично авторизований до ролей, що нижчі за ієрархією.

Важливим у моделі РРД є також обмеження на множину ролей.

Означення В базовій моделі РРД задано обмеження статичного взаємного виключення ролей, чи прав доступу, якщо виконуються умови:

$$R = R_1 \cup ... \cup R_n, R_i \cap R_j = \emptyset$$

$$\left|UA(u) \cap \ _j\right| \leq \ ,i = \overline{,n}$$

$$P = P_1 \cup ... \cup P_n, P_i \cap P_j = \emptyset$$

$$\left|PA(u) \cap \ \right| \leq \ ,i = \overline{,n}$$

Означення В базовій моделі РРД задано обмеження динамічного взаємного виключення ролей, чи прав доступу, якщо виконуються умови:

$$R = \ _1 \cup \ \cup \ _n, R_i \cap \ _j = \emptyset \ \leq \ \leq \ \leq \ ^{\cdot} \leq \ ;$$
$$\left|roles(s) \cap \ _i\right| \leq \ ,s \in \ ^{\prime},i - \ 2,...,n.$$

Тобто в кожній сесії користувач може володіти не більше ніж однією роллю із кожної підмножини ролей.

Означення В базовій моделі РРД задано статистичні кількісні обмеження на володіння роллю чи доступом, якщо визначенні дві функції:

$$\alpha \ \ R \rightarrow \ _{,};$$
$$\beta \ \ P \rightarrow \ _{,}$$

N_0 **-** множина натуральних чисел з нулем**,** де

$$\left|UA^{-}(r)\right| \leq \alpha \ ), r \in \ ^{\cdot};$$
$$\left|PA^{-}(p)\right| \leq \beta \ p), p \in \ ^{\cdot}.$$

Для кожної ролі встановлюється максимальна кількість користувачів , а для кожного доступу визначається максимальна кількість ролей.

Означення В базовій моделі РРД задано динамічно кількісні обмеження на володіння роллю чи доступом, якщо визначенні дві функції:

$\gamma : R \to$ з умовою $\left|roles^{-1}(r)\right| \le \gamma\ r), r \in$.

Для ролі встановлюється максимальна кількість сесій користувача, які можуть одночасно авторизуватись на цю роль.

Означення В базовій моделі РРД задано статистичне обмеження необхідного володіння роллю чи правом доступу, якщо визначено дві функції

$\alpha \quad R \to$;

$\beta \quad P \to$

і виконуються умови

$для\ u \in$, $якщо\ r, r' \in$, $r \in$ $A(u), r' \in \alpha$), $то\ r' \in$ $A(u)$.

$для\ r \in$, $якщо\ p, p' \in$, $p \in$ $A(r), p' \in \beta$ $p), то\ p' \in$ $A(r)$

Для кожної ролі повинно бути вказано список додаткових ролей на які вже повинен бути авторизований користувач перш ніж авторизуватись на дану роль, для кожного права доступу перш ніж його отримати роль повинна володіти вже деякою множиною доступів.

Означення В базовій моделі РРД задано динамічне обмеження необхідного володіння роллю чи правом доступу, якщо визначено функцію:

$\gamma : R \to$

і виконується умова

$для\ s \in$, $якщо\ r, r' \in$, $r \in$ $les(s), r' \in \gamma$ $r), то\ r' \in$ $les(s)$

Для кожної ролі для того, щоб на неї міг бути авторизований користувач в певній сесії повинні бути визначенні ролі на які користувач також повинен авторизовуватись.

Перелік знань та умінь, якими повинен оволодіти студент.

У процесі виконання лабораторної роботи студент повинен знати:

- призначення та структура рольового управління доступом;
- проблеми моделювання рольового управління доступом;

- вимоги до безпеки системи захисту побудованої на основі рольового управління доступом;

- складності у реалізації рольових моделей розмежування доступу та сфери їх застосування;

- взаємозв'язок рольової політики безпеки із мандатної та дискреційною політикою;

- перелік моделей, що реалізують рольову політику безпеки.

Після виконання лабораторної роботи студент повинен вміти:

- будувати алгоритми для програмної реалізації рольових моделей управління доступом;

- програмно реалізовувати моделі, що реалізують рольову політику безпеки;

- складати технічне завдання на програмну розробку;

- реалізовувати програмні додатки, що можуть працювати в різних сеансах;

- реалізовувати алгоритми розподілу та надання ролей користувачам;

- формувати звіти.

Вимоги до програми лабораторної роботи.

Порядок виконання роботи

Постановка завдання: створити операційну оболонку, в якій реалізовано рольове управління доступом та реалізуються наступні функції:

- реєстрація та ідентифікація суб'єктів;
- реєстрація користувачів;
- створення файлів
- створення модуля контролю сеансів;
- надання постійних ролей користувачу та ролей відповідно до сеансу;
- надання постійного доступу ролям та доступу в сеансі;
- регулювання доступу за допомогою монітора безпеки згідно моделі;
- вести журнал аудиту;
- реалізувати в системі статичне та динамічне взаємне виключення ролей.

Етапи виконання:

1. Ознайомитись із теоретичними положеннями моделі рольового розмежування доступу;
2. Вибрати середовище програмування та обґрунтувати вибір;
3. Створення модуля реєстрації та ідентифікації користувачів та об’єктів;
4. Створення модуля контролю сеансів;
5. Створення модуля присвоєння ролей;
6. Створення модуля розмежування доступу для ролей;
7. Створення модуля адміністратора;
8. Створення модуля перевірки права на доступ, який хоче отримати користувач в реальному режимі (монітор безпеки);
9. Створення модуля аудиту подій доступу із можливістю вибрати тип подій, які необхідно протоколювати.

Бажаний результат роботи

Результатом виконання лабораторної роботи є:

- закінчений програмний продукт (операційна оболонка), що реалізує всі зазначенні у постановці завдання функції;
- звіт, що подається у електронному та друкованому вигляді (порядок оформлення звіті нижче по тексту);
- лістинг програмного продукту із коментарями, що роздруковується після успішного запуску програми на виконання.

Форма звітності.

Звіт повинен подаватись в електронному та друкованому вигляді і повинен містити наступні розділи:

- короткий опис проблем програмної реалізації вибраної моделі;
- обґрунтування вибору середовища програмування;
- специфікація програмного продукту:

- призначення розробки;

- вимоги до програмного продукту:

- *вимоги до функціональних характеристик.*

- *умови експлуатації.*

- *вимоги до інформаційної та програмної сумісності.*

- техніко-економічні показники програми;

- керівництво користувача;

- тестування програми;

- висновок.

Варіанти завдань.

При реалізації операційної оболонки за основу слід вибрати одну із запропонованих моделей рольового управління доступом згідно отриманих варіантів.

Варіант 1. Базова модель рольового розмежування доступу.

Варіант 2. Адміністративна модель рольового розмежування доступу.

Варіант 3.Мандатна модель рольового розмежування доступу.

Варіант 4. Дискреційна модель рольового розмежування доступу.

Варіант 5. Модель ядра безпеки на основі рольового розмежування доступу.

Список джерел

1. Щеглов Ю.А. Защита компьютерной информации от несанкционированного доступа. – СПб: Наука и Техника, 2004. – 384 с.
2. Девянин П.Н Модели безопасности компьютерных систем: Учебн. пособие для студентов высш. учеб. завед. – М.: Издательский центр «Академия», 2005. – 144 с.
3. Домарев В. В. Безопасность информационных технологий. Методы создания систем защиты. — К.: ООО ТИД ДС,2001. — 688 с.
4. Руководство к практическим занятиям по курсу Теоретические основы информационной безопасности. Под ред. Зегжды П.Д. СПб: Изд-во СПбГТУ, 1998. – 20 с.
5. Калинин М.О. Язык описания политик безопасности информационных систем/ Современное машиностроение: Сб. трудов молодых ученых. Вып. 1. — СПб: Изд-во СПбИМаш, 1999. — с. 69-74.

6. Карминский А.М., С.А. Карминский, В.П. Нестеров, Б.В. Черников. Информатизация бизнеса: концепции, технологии, системы.- М.: Финансы и статистика, 2004 г. – 624 с.

Лабораторна робота № 4.

Тема роботи: проектування та оцінка системи захисту на основі ймовірнісних моделей.

Мета роботи: програмно реалізувати алгоритм оцінки та проектування системи захисту на основі ймовірностей моделі.

Короткі теоретичні відомості.

Основою формального опису систем захисту з метою їх оцінки традиційно вважається модель системи захисту з повним перекриттям, у якій розглядається взаємодія "області загроз", "області, що захищається" і "системи захисту" (механізмів безпеки).

Таким чином, маємо три множини, які опишемо в термінах загальної моделі функціонування СЗ:

$P^{(s)} = \{p_i\}$ – множина загроз безпеці,

$R^{(c)} = \{r_j\}$ – множина об’єктів захищеної системи,

$S^{(o)} = \{s_k\}$ – множина механізмів безпеки.

Елементи цих множин знаходяться між собою у визначених відносинах, що і описують систему захисту.

Для опису системи захисту звичайно використовується графова модель, представлена на рис 4.1. Множина відносин загроза-об'єкт утворює дводольний граф $\langle P^{(s)}, R^{(c)} \rangle$. Ціль захисту полягає в перекритті всіх можливих ребер в графі. Це досягається введенням третього набору $S^{(o)}$. У результаті маємо тридольний граф $\langle P^{(s)}, S^{(o)}, R^{(c)} \rangle$.

Розвиток цієї моделі припускає введення ще двох елементів.

V - набір вразливих місць, обумовлений підмножиною декартового добутку: $P^{(s)} * R^{(c)}$ $v_r = \langle p_i, r_j \rangle$.

Отож в моделі з повним перекриттям під вразливістю системи захисту будемо розуміти можливість здійснення погрози p у відношенні об'єкта r.

B - набір бар'єрів, обумовлений декартовим добутком $V * S^{(o)} : b_l = \langle r_i, p_j, s_k \rangle$, що представляють собою шляхи реалізації загроз безпеці, перекриті засобами захисту.

У результаті одержуємо систему, що складається з п'яти елементів: $\langle P^{(s)}, S^{(o)}, R^{(c)}, V, B \rangle$ (Рис.4.2)

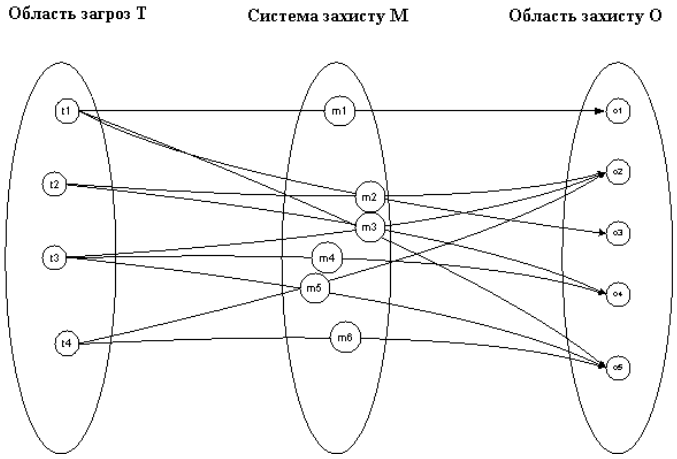


Рисунок 4.1 *Графова модель системи захисту з повним перекриттям.*

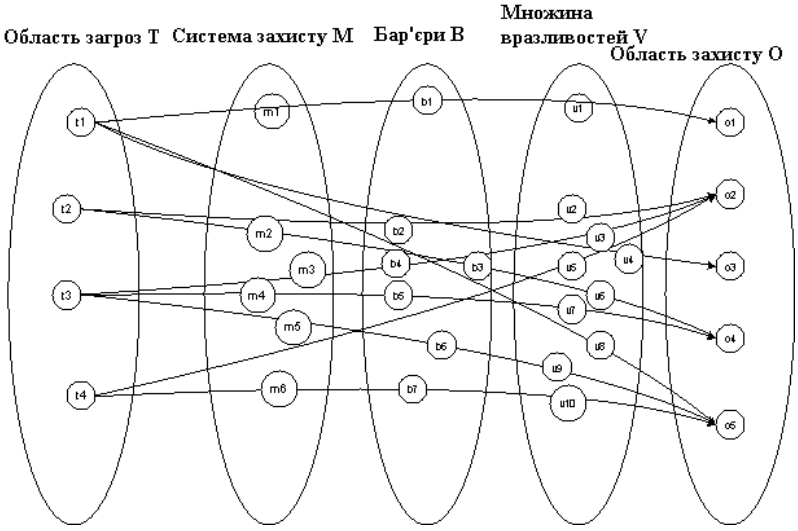


Рис. 4.2 *Модель системи захисту із врахуванням вразливостей.*

В даній моделі показник захищеності K є функцією вигляд якої залежить від вхідних параметрів. Нехай маємо:

P_l - імовірність появи загрози;

Q_l - величина збитку при реалізації загрози;

R_l - ступінь спротиву механізму захисту.

Визначення захищеності по моделі з повним перекриттям залежить від міцності бар'єра b_l, що характеризується величиною залишкового ризику $Risk_l$, пов'язаного з можливістю здійснення загрози p_i у відношенні об'єкта o_j, при використанні механізму захисту m_k. Ця величина визначається по формулі:

$$Risk_l = P_k Q_k (1 - R_k) \qquad (4.1)$$

Тоді враховуючи (4.1) для визначення величини захищеності K можна використовувати наступну формулу:

$$K = 1 \Big/ \sum_{\forall л} P_k Q_k (1 - R_k),$$

де $P_k, Q_k \in [0,1]$ $R_k \in [0,1]$.

Слід зауважити, що ця формула має зміст лише тоді виконується умова, що для будь-якої вразливості є відповідний бар'єр, що усуває цю вразливість.

Для оцінки СЗ за графічною моделлю слід зробити наступні припущення:

- всі елементи ІС потребують однакового захисту;

- забезпечення захисту кожного із елементів ІС є подія незалежна.

Дана модель дозволяє описувати процес захисту ІС, які можна розділити на класи за рівнем цінності активів.

Нехай СЗ спроектована на основі стандарту. Позначимо надійність захисту i-елементу через e_i, тоді, враховуючи, що події забезпечення захисту елементів ІС незалежні, загальна ефективність захисту буде рівна

$$E = \prod_{i=1}^{n} e_i \qquad (4.2)$$

У вірогідносному просторі e_i вираз (4.2) визначає гіперболічну гіперповерхню із асимптотичною гіперповерхнею, що являє собою круговий конус, тому максимальне значення цей вираз приймає асимптотичній гіперповерхні, тобто при умові, що $e_1 = \ldots = e_n$. Початкова задача оцінки СЗ формулюється наступним чином: необхідно забезпечити максимальну ефективність СЗ при заданих витратах S. Припускається, що $e_i = f_i(S_i)$, де S_i-

частина ресурсу S. Оскільки (4.2) досягає максимуму при рівності $e_1 = \ldots = e_n$, то задача приймає наступний вигляд

$$f_1(S_1) = \ldots = f_n(S_n) \to \max$$

$$\sum_{i=1} S_i = S,$$

$$e_i \geq A, i = \overline{1,n}$$

де A-нижня межа допустимого рівня гарантованої оцінки.

Для використання даної моделі слід зазначити, що функції $e_i = f_i(S_i)$ повинні бути монотонно зростаючі в строгому змісті.

Якщо функції $e_i = f_i(S_i)$ та $S_i = f^{-1}(e_i)$ можна задати графічно то дана задача розв'язується графічним методом.

Приклад.

Припустимо, що ефективність захисту в залежності від рівня вкладень підпорядковується нормальному закону розподілу. Тоді ймовірності досягнення ефективністю захисту деякого порога A буде визначатися формулою

$$e(A \setminus \sigma) = 2/\sqrt{2\pi} \int_0^{A/\sigma} e^{-s^2/2}\, ds \qquad (4.3)$$

Дисперсії $\sigma_1, \sigma_2, \sigma_3, \sigma_4$, що відповідають оцінкам ефективності відповідно цілісності, конфіденційності, доступності, спостережності, визначаються експертним шляхом. Графіки функція для відповідних загроз при умові, що $\sigma_1 > \sigma_2 > \sigma_3 > \sigma_4$ будуть мати наступний вигляд (рис. 1.)

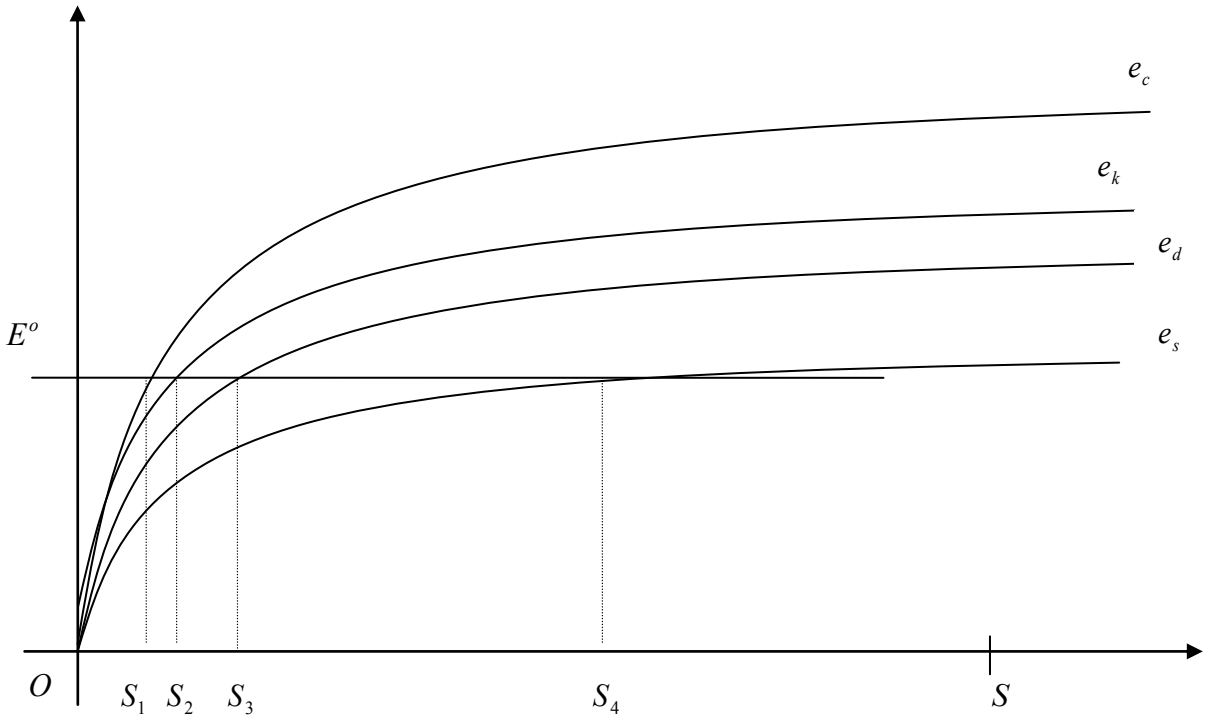


Рис. 4.3. Розв'язок задачі (4.3) графічним методом

Як видно на рис 4.3. розв'язати задачу (4.3) означає провести найвищу з можливих ліній паралельних осі OS, такої, щоб задовольнялась умова $\sum_{i=1}^{4} S_i = S$

Рубіжна модель оцінки систем захисту.

Рубіжна модель базується на припущені, що СЗ можна подолати чи обійти. Тоді захищеність визначається, як

$$K = 1 - P_{pd})(1 - P_{ob}), \qquad (4.4)$$

де K - імовірність захищеності, P_{pd} - імовірність подолання СЗ, P_{ob} - імовірність обходу СЗ та при умові, що час подолання, чи час обходу набагато менші періоду цінності інформації.

В свою чергу

$$P_{pd} = 1 - \text{'}_{pd} / t_{ko}, \qquad (4.5)$$

де t_{pd} - час подолання СЗ, t_{kn} - час контролю за НСД. Зазначимо, що при $t_{pd} > {}_{kn}$, то $P_{pd} =)$.

Тоді імовірність обходу буде обчислюватись за формулою:

$$P_{ob} = 1 - t_{pd} / t_{ob}, \qquad (4.6)$$

де t_{ob} - час обходу СЗ.

Тоді враховуючи (4.4) – (4.6) захищеність СЗ будемо визначати, як:

$$K = \sum_{\eta} C_{\eta} (1 - t_{pd} / t_{ko}) \min_{\forall b} (1 - t_{pd} / t_{ob}),$$

де $C_{\eta} = \frac{\Delta_{\eta}}{\Delta}$ - період перебування системи в η - тому стані, а мінімум визначається у тому випадку, якщо є декілька способів обходу СЗ.

Модель оцінки системи захисту на основі теорії математичного моделювання.

Степінь захищеності *К* будемо визначати із такої залежності:

$$K = f(C, p, E, Q),$$

де C - вартість інформації, що захищається;

p - ймовірність взлому;

E - вартість СЗ;

Q - продуктивність ІС.

Спираючись на вищесказане критерії оцінки СЗ можна подати у вигляді схеми (рис 1.9)

Як видно із схеми загальну модель оцінки можна подати у вигляді однокретеріальної задачі математичного програмування із двома обмеженнями :

$$\begin{cases} D(C, p) \to \max \\ E_{сз} \le E_{op} \\ Q_{сз} \le Q_{op} \end{cases} \qquad (4.7)$$

де $E_{сз}$ - вартість СЗ, E_{op} - задане обмеження на вартість СЗ, $Q_{сз}$ - зниження продуктивності ІС після встановлення СЗ, Q_{op} - задані обмеження на зменшення продуктивності.

Коефіцієнт захищеності вводиться через параметри загроз. Введемо наступні позначення [1]:

w - кількість видів загроз, що впливають на систему;

C_i $i = \overline{,w}$ - вартість втрат від загрози i – того виду;

λ $(= \overline{,w}$ - інтенсивність потоку загроз i – того виду;

S_i $(= \overline{,w}$ - ймовірність появи загрози i – того виду в загальному потоці загроз, $S_i = \frac{\lambda}{\Lambda}$;

$p_i(i = \overline{,w})$ - імовірність попередження впливу i-тої загрози. Тоді рівень захищеності D буде визначатись за формулою:

$$D = 1 - \frac{\sum_{i=1}^{w} C_i \cdot S_i (1 - p_i)}{\sum_{i=1}^{w} C_i \cdot S_i} = 1 - \frac{\sum_{i=1}^{w} C_i \cdot \lambda (1 - p_i)}{\sum_{i=1}^{w} C_i \cdot \lambda}.$$

Якщо розрахований коефіцієнт захищеності не влаштовує, тоді змінюються обмеження в межах допустимих норм та знову проходить обчислення.

Однією з основних проблем даних моделей є методи отримання вхідних параметрів та отримання статистичних даних. Для отримання таких результатів відносно імовірнісних характеристик загроз слід застосувати відомі моделі атак.

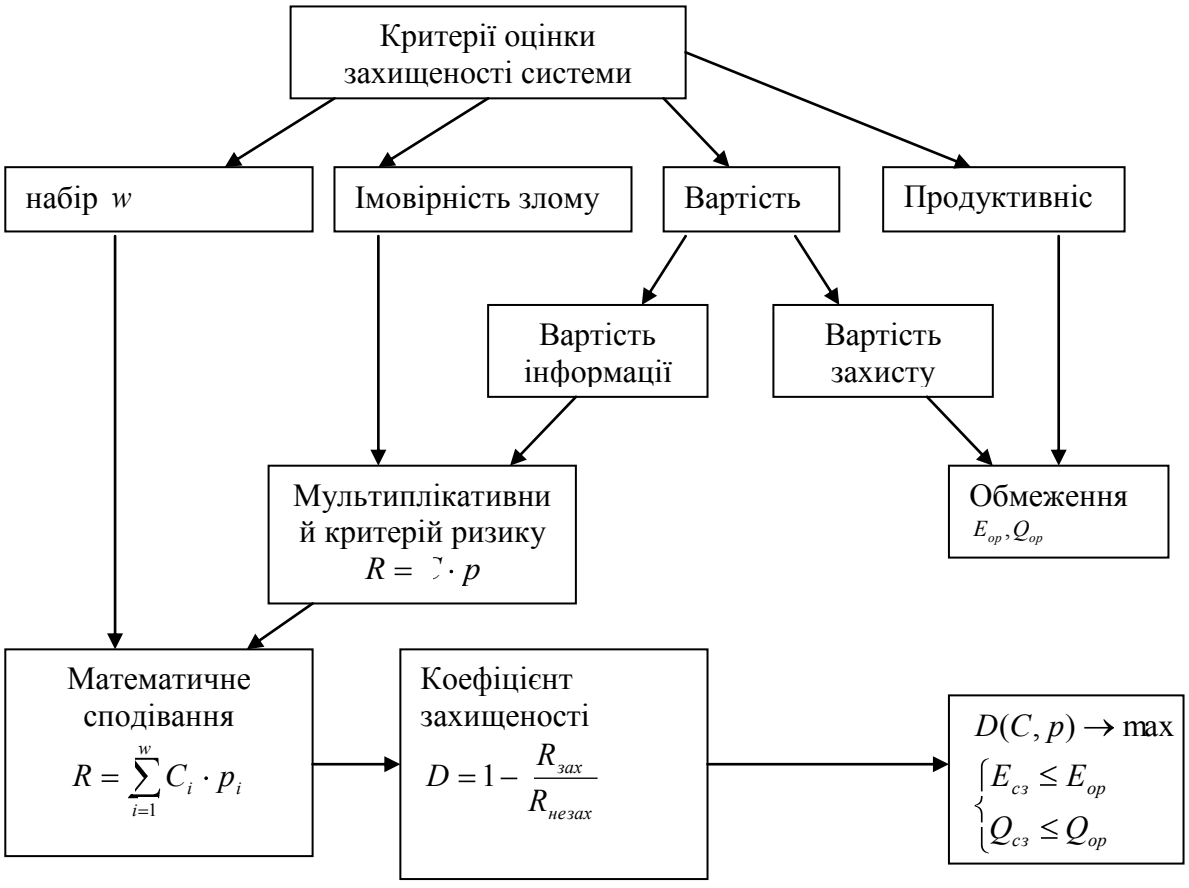


Рис. 4.4 *Схема визначення коефіцієнта захищеності.*

Перелік знань та умінь, якими повинен оволодіти студент.

У процесі виконання лабораторної роботи студент повинен **знати:**

- етапи проектування систем захисту інформації;

- поняття рівня захищеності інформаційної системи та шкали і одиниці його вимірювання;

- класифікацію моделей оцінки захищеності інформаційних систем;

- основні поняття теорії графів, теорії ймовірностей, математичного програмування;

- методики оцінки систем захисту згідно вибраної моделі.

Після виконання лабораторної роботи студент повинен **вміти**:

- будувати алгоритм для програмної реалізації ймовірнісних моделей оцінки захищеності інформаційних систем;

- складати технічне завдання на програмну розробку;

- використовувати середовище програмування для реалізації алгоритму;

- проектувати та реалізовувати модулі інтерактивного опитування користувача та опрацювання відповідей;

- застосовувати розроблений програмний засіб для оцінки рівня захищеності конкретної ІС;

- працювати із власними типами даних при опрацювання параметрів із великими числовими значеннями;

- формувати звіти.

Вимоги до програми лабораторної роботи.

Порядок виконання роботи

Постановка завдання: створити прикладну програму, яка призначена для оцінки систем захисту згідно наперед вибраної моделі та яка реалізує наступні функції:

- реєстрація та опис ІС, як об’єкту оцінки захищеності;
- введення даних необхідних для оцінки захищеності ІС згідно вибраної моделі;
- збір системних даних про деякі параметри ІС автоматично;

- проведення інтерактивного опитування користувачів для отримання необхідних параметрі для оцінки;
- оцінка рівня захищеності ІС згідно наперед вибраної моделі;
- формування звіту щодо результатів оцінки інформаційної системи

Етапи виконання:

1. Ознайомитись із теоретичними положеннями ймовірнісних моделей оцінки рівня захищеності інформаційної системи;
2. Вибрати середовище програмування та обґрунтувати вибір;
3. Розробка та реалізація модуля реєстрації об’єктів оцінки захищеності;
4. Розробка та реалізація модуля внесення необхідних параметрів для оцінки захищеності ІС;
5. Розробка та реалізація модуля автоматичного збору даних про ІС, якщо на це отримано відповідний дозвіл;
6. Розробка та реалізація модуля оцінки захищеності ІС згідно вибраної моделі.
7. Розробка та реалізація модуля формування звіту про результати оцінки та можливі рекомендації по підвищенню рівня захищеності.

Бажаний результат роботи

Результатом виконання лабораторної роботи є:

- закінчений програмний продукт, що реалізує всі зазначенні у постановці завдання функції;
- звіт, що подається у електронному та друкованому вигляді (порядок оформлення звіті нижче по тексту);
- лістинг програмного продукту із коментарями, що роздруковується після успішного запуску програми на виконання.

Форма звітності.

Звіт повинен подаватись в електронному та друкованому вигляді і повинен містити наступні розділи:

- короткий опис проблем програмної реалізації вибраної моделі;
- обґрунтування вибору середовища програмування;

- специфікація програмного продукту:

- призначення розробки;

- вимоги до програмного продукту:

- *вимоги до функціональних характеристик.*

- *умови експлуатації.*

- *вимоги до інформаційної та програмної сумісності.*

- техніко-економічні показники програми;

- керівництво користувача;

- тестування програми;

- висновок.

Варіанти завдань.

При реалізації програмного додатку за основу слід вибрати одну із запропонованих ймовірнісних моделей оцінки захищеності ІС згідно отриманих варіантів.

Варіант 1. Модель «повного перекриття».

Варіант 2. Розширена модель «повного перекриття».

*Варіант 3.*Рубіжна модель оцінки системи захисту інформації.

Варіант 4. Графічна модель оцінки систем захисту інформації.

Варіант 5. Модель оцінки захищеності на основі теорії математичного програмування.

Варіант 6. Ігрова модель оцінки систем захисту інформації.

Список джерел

1. Щеглов Ю.А. Защита компьютерной информации от несанкционированного доступа. – СПб: Наука и Техника, 2004. – 384 с.
2. Девянин П.Н Модели безопасности компьютерных систем: Учебн. пособие для студентов высш. учеб. завед. – М.: Издательский центр «Академия», 2005. – 144 с.
3. Комплексная безопасность объекта: от теории к практике/ С.М. Доценко, В.Ф. Шпак. – С.-Петербург: ООО. Изд. «Полигон» – 2000. – 300 с.

4. Конахович Г. Ф. Защита информации в телекоммуникационных системах. МК «Пресс», Киев, 2005. – 260 с.
5. НД ТЗІ 2.5-004-99. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу.
6. Ларс Кландер. Hacker Proof: Полное руководство по безопасности компьютера. – Минск: «Попурри».-2002.- 688 с.

Лабораторна робота № 5.

Тема роботи: Реалізація математичної моделі комп'ютерної атаки.

Мета роботи: програмно реалізувати одну із моделей комп'ютерної атаки, проаналізувати недоліки та переваги, що притаманні подібним моделям.

Короткі теоретичні відомості.

Атака – це сукупність дій порушника, що направлені на реалізацію загрози безпеці ІС, шляхом активації тої чи іншої вразливості ІС.

При конкретизації типу ІС, воно може бути уточнене. Зокрема, розглядаючи ІС, як комп'ютерну мережу (КС), атаку можна визначити, як несанкціонований інформаційний вплив на об'єкти мережі, що здійснюються, як по каналам зв'язку, так і при безпосередньому контакті.

Функціональна модель атаки.

Парр Г.Л., Марков А.С. пропонують підхід, що базується на представлені атак, як сукупності реалізацій спланованих загроз СЗ – подій чи дій, що можуть суттєво знизити рівень безпеки ІС.

Згідно [3], ці загрози розділяються на три класи:

- операційні дефекти;
- дефекти адміністрування;
- дефекти користувача.

Операційні дефекти характеризують технологічну безпеку і з'являються на етапі проектування та реалізації СЗ. Основними типами операційних дефектів є:

- недоліки механізму автентифікації;
- недоліки механізму розмежування прав;
- недоліки механізму цілісності;
- недоліки механізму криптографії;
- недоліки криптографічного протоколу;
- недоліки мережевих протоколів;
- помилки програмної реалізації.

Помилки адміністрування показують проблеми безпеки на експлуатаційному етапі, які спричиняються невідповідністю налаштувань СЗ і ПБ.

Основними типами помилок адміністрування є:

- помилки параметрів підключення користувачів;
- помилки налаштування парольного захисту, використання паролів, які легко підбираються
- помилки конфігурації сервера, призначення привілегій.

Спираючись на досвід варто виділити помилки користувачів, адже хоча користувачі не приймають участі в адмініструванні та налаштовуванні СЗ, та їхні навмисні чи ненавмисні дії можуть нести загрозу ІБ.

Виходячи з цього пропонується така функціональна модель атаки:

$A = \{O, T_d, T_a, T_u, R_t, R_{ot}, K\}$,

де О – множина об'єктів, що захищаються (цільова множина атак);

T_d - множина операційних дефектів;

T_a - множина помилок адміністрування;

T_u - множина помилок користувача;

$R_t = T * T$ - декартова множина загроз ($T = \{T_d, T_a, T_u\}$);

$R_{ot} = R_t * O$ - декартова множина загроз та об'єктів;

K - множина об'єктів СЗ.

Ця модель має описовий характер і дозволяє формально представити атаку з точки зору її мети, та використовуваних нею вразливостей. Позитивна сторона цієї моделі полягає в тому, що вона дозволяє чітко розділити області впливу атаки, розглянути варіанти використання тої чи іншої області дефектів.

Концептуальна модель атаки.

Аналізуючи атаки з точки зору планування Городецкий В.И., Котенко И.В [29] зауважують, що кожна атака попередньо планується на *макрорівні* у виді частково впорядкованої множини кроків, що складають разом сценарій атаки.

Наприклад типовим сценарієм реалізації атак, як показують Медведовский И.Д., Семьянов П.В., Леонов Д.Г є:

1) аналіз мережевого трафіка;

2) сканування мережі;

3) підміну довіреного об'єкта мережі і передачу по каналах зв'язку повідомлень від його імені з присвоєнням його прав доступу;

4) впровадження хибного об'єкта в мережу;

5) відмова в обслуговуванні;

6) неавторизований доступ з віддаленого хосту за допомогою підбору пароля;

7) неавторизоване підвищення привілеїв доступу;

8) віддалений запуск додатків.

Кожен крок атаки спрямований на досягнення часткової мети (наприклад, аналіз комп'ютерної мережі, що атакується, подолання системи автентифікації, підвищення прав, одержання доступу до інформації, “замітання слідів”). Ці кроки можуть бути реалізовані в різному, хоча і не в довільному порядку, повторюватися і виконуватися з різних віддалених комп'ютерів. Кожен крок сценарію атаки реалізується послідовністю простих команд і операцій *мікрорівня*. Спираючись на таке представлення атак, пропонується дворівнева концептуальна модель атак.

На першому макрорівні задається загальний сценарій атаки. Навіть одиничний прецедент такого сценарію дозволяє експерту ідентифікувати намір особи, що атакує, особливості виконання атаки, її варіанти і змінні параметри, що ведуть до тієї ж самої мети. Кожен сценарій описується множиною припустимих послідовностей кроків, що визначають клас атак на макрорівні. Другий мікрорівень визначає більш детальну специфікацію атаки. Кожен крок сценарію макрорівня на мікрорівні складається з послідовності подій. Цими подіями є конкретні команди операційної системи, що викликаються стандартними додатками і експлоїти з конкретними параметрами виклику.

Деревовидна модель атаки.

Andrew P. Moore, Robert J. Ellison, Richard C. Linger в своїй статті Attack Modeling for Information Security Technical Note [42] вказують, що для побудови моделі атаки її потрібно документувати в структурованій формі. Алгоритм побудови моделі такий:

1) будуються палітри (patterns) типових атак;

2) будуються дерева (trees) типових атак;

3) із дерева виділяються всі можливі сценарії атак.

Дерево сценаріїв складається із вузлів, що характеризують цілі, чи підцілі атак. Вузол дерева складається із:

1) набору підцілей атаки кожна з яких має бути досягнута для успішного здійснення атаки (I - декомпозиція);

2) набору підцілей, з яких хоч одна має бути досягнута для успішного здійснення атаки (АБО - декомпозиція)

Покажемо графічне представлення цих декомпозицій.

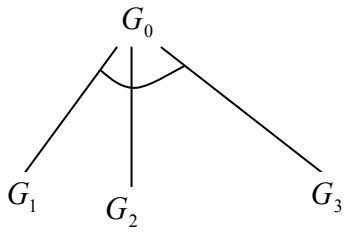


Рис.5.1 I - декомпозиція

Ціль G_0 буде досягнуто лише в тому випадку коли будуть досягнуті підцілі G_1, G_2, G_3.

Ціль G_0 буде досягнуто коли буде досягнута хоч одна з підцілей G_1, G_2, G_3.

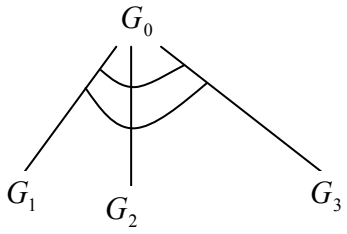


Рис.5.2 АБО - декомпозиція

Дерево атаки може складатись із будь – якого набору І - , чи АБО – декомпозицій. Кожний сценарій атаки генерується обходом дерева від кореня до листя.

Приведемо простий приклад дерева атак та виділимо із нього сценарії (Рис. 5).

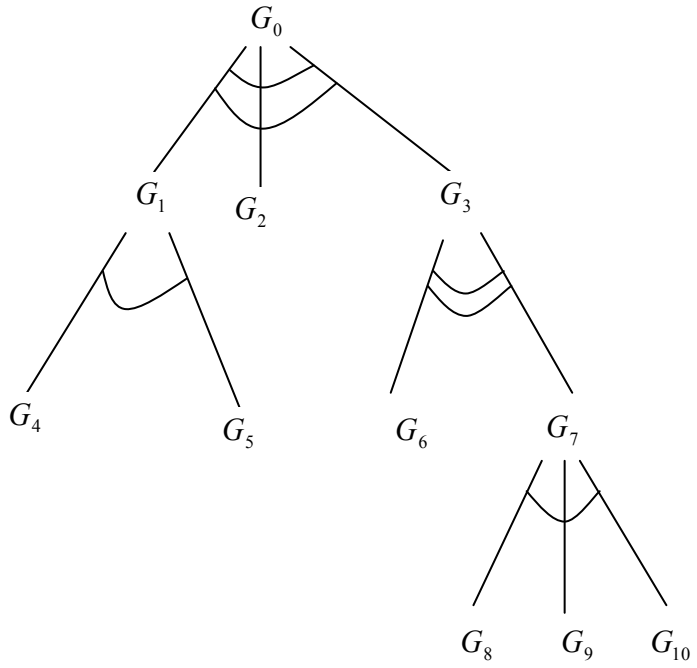


Рис. 5.3 Дерево атаки на ціль G_0

По даному дереву можна згенерувати такі сценарії атак на G_0:

$\{G_0, G_1, (G_4, G_5)\}$
$\{G_0, G_2\}$
$\{G_0, G_3, G_6\}$
$\{G_0, G_3, G_7, (G_8, G_9, G_{10})\}$

Визначення цілі атаки дає змогу об’єкти ІС та СЗ, які можуть бути скомпрометовані. Зазначимо, що саме ця модель дає найкращий механізм для

оцінки впливу атаки. Якщо припустити, що за деяким методом обчислення цінність кожного об'єкт, тоді кожній цілі, чи підцілі атаки можна надати *ваговий коефіцієнт*, що дорівнює цінності об'єкта на який вони впливають. Звідси загальний збиток, що може завдати атака, визначається як сума вагових коефіцієнтів, кожної декомпозиції.

Нехай підціль G_i має ваговий коефіцієнт $V_j, j = \overline{,n}$. Ваговий коефіцієнт I – декомпозиції буде обчислюватись, як

$$V_I = \sum_{j=1}^{k} V_j ,$$

де k - кількість підцілей, що входять в I – декомпозицію, а V_j їх відповідні вагові коефіцієнти.

Для АБО – декомпозиції обчислення вагового коефіцієнта буде здійснюватись за формулою

$$V_{АБО} = \max_{j=\overline{,k}} V_j ,$$

де k - кількість підцілей, що входять в АБО – декомпозицію, а V_j їх відповідні вагові коефіцієнти.

Тоді потенційний загальний збиток, що може завдати атака буде визначатись, як

$$V = V_I + V_{АБО},$$

де k,l - кількість І- та АБО – декомпозицій відповідно.

Зауважимо, що глибина декомпозиції визначається глибиною оцінки об'єктів. Тобто, якщо за певною методикою оцінюється кожен окремий об'єкт, то і декомпозиції підцілей має бути до рівня конкретних об'єктів.

Перелік знань та умінь, якими повинен оволодіти студент.

У процесі виконання лабораторної роботи студент повинен **знати:**

- визначення та класифікація комп'ютерних атак;
- етапи реалізації комп'ютерної атаки;
- класифікацію моделей комп'ютерних атак;
- основні поняття теорії графів, теорії ймовірностей;

- методики аналізу комп’ютерних атак.

Після виконання лабораторної роботи студент повинен **вміти**:

- будувати алгоритм для програмної реалізації моделей оцінки комп’ютерних атак;

- складати технічне завдання на програмну розробку;

- використовувати середовище програмування для реалізації алгоритму;

- описати комп’ютерну атаку в рамках вибраної моделі;

- застосовувати розроблений програмний засіб для оцінки рівня збитку ІС від реалізації атаки, що моделюється;

- подавати рекомендації по запобіганню певних видів атак;

- формувати звіти.

Вимоги до програми лабораторної роботи.

Порядок виконання роботи

Постановка завдання: створити прикладну програму, яка призначена для моделювання атаки на комп’ютерну систему згідно наперед вибраної моделі та яка реалізує наступні функції:

- створення шаблонів атак згідно вибраної моделі;
- введення даних необхідних для моделювання конкретної атаки;
- опис інформаційної системи, що потенційно може зазнати атаки;
- виведення результатів що моделюють вплив атаки на описану ІС;
- формування рекомендацій про вдосконалення ІС для захисту від вибраного типу атак;
- аналіз реалізації впливу атак залежно від виду управління доступом в ІС.

Етапи виконання:

1. Ознайомитись із теоретичними положеннями моделювання атак на комп’ютерні системи;
2. Вибрати середовище програмування та обґрунтувати вибір;
3. Розробка та реалізація модуля створення шаблонів атак згідно вибраної моделі;

4. Розробка та реалізація модуля внесення параметрів для створення шаблону конкретної атаки;
5. Розробка та реалізація модуля опису ІС;
6. Розробка та реалізація модуля співставлення параметрів атаки та ІС та опису результатів.
7. Розробка та реалізація модуля формування звіту по вдосконаленню ІС.

Бажаний результат роботи

Результатом виконання лабораторної роботи є:

- закінчений програмний продукт, що реалізує всі зазначенні у постановці завдання функції;
- звіт, що подається у електронному та друкованому вигляді (порядок оформлення звіті нижче по тексту);
- лістинг програмного продукту із коментарями, що роздруковується після успішного запуску програми на виконання.

Форма звітності.

Звіт повинен подаватись в електронному та друкованому вигляді і повинен містити наступні розділи:

- короткий опис проблем програмної реалізації вибраної моделі;
- обґрунтування вибору середовища програмування;
- специфікація програмного продукту:

- призначення розробки;
- вимоги до програмного продукту:
- *вимоги до функціональних характеристик.*
- *умови експлуатації.*
- *вимоги до інформаційної та програмної сумісності.*
- техніко-економічні показники програми;
- керівництво користувача;
- тестування програми;
- висновок.

Варіанти завдань.

При реалізації програмного додатку за основу слід вибрати одну із запропонованих моделей комп'ютерних атак.

Варіант 1. Функціональна модель атак.

Варіант 2. Концептуальна модель атак.

Варіант 3. Деревовидна модель атак.

Варіант 4. Метод моделювання атак на основі ймовірнісних автоматів.

*Варіант 5.*Модель розповсюдження атак на основі виміру довжини гамільтонового шляху.

Варіант 6. Модель різницевих атак на криптосистеми.

Список джерел

1. Щеглов Ю.А. Защита компьютерной информации от несанкционированного доступа. – СПб: Наука и Техника, 2004. – 384 с.
2. Теоретические основы компютерной безопасности. / П.Н. Девянин, О.О. Михальский, Д.И Правиков и др. – М.: Радио и связь, 2000. – 192 с.
3. Девянин П.Н Модели безопасности компьютерных систем: Учебн. пособие для студентов высш. учеб. завед. – М.: Издательский центр «Академия», 2005. – 144 с.
4. Куприянов А.И. Основы защиты информации.- М.: Издательский центр «Академия», 2006. – 256 с.
5. Малюк А.А. Информационная безопасность:концептуальные и методологические основы защиты информации. Учеб. Пособие для вузов.- М.: Горячая линия. – Телеком, 2004 .- 208 с.
6. Запечников С. В., Милославская Н. Г., Толстой А. И., Ушаков Д. В. И74 Информационная безопасность открытых систем: Учебник для вузов. В 2-х томах. Том 2 – Средства защиты в сетях / – М.: Горячая линия–Телеком, 2008. – 558 с.
7. Мещеряков Р.В., Праскурин Г.А. Теоретические основы компьютерной безопасности. Курс лекций. Раздел 1. Томск: Изд-во ТУСУР, 2005. – 147 с.