МІНІСТЕРСТВО ОБОРОНИ УКРАЇНИ
Військовий інститут телекомунікацій та інформатизації
Національного технічного університету України
„Київський політехнічний інститут”

ВОЄННО-НАУКОВА КОНФЕРЕНЦІЯ КУРСАНТІВ (СТУДЕНТІВ)

Факультету інформаційних технологій в системах управління
23 травня 2013 року

(Доповіді та тези доповідей)

Київ – 2013

ББК
Ц4 (4Укр)39
П-768

У збірнику матеріалів воєнно-наукової конференції курсантів (студентів) факультету інформаційних технологій у системах управління опубліковано доповіді та тези доповідей вчених, науково-педагогічних працівників, ад'юнктів, здобувачів, курсантів і студентів Військового інституту телекомунікацій та інформатизації Національного технічного університету України „Київський політехнічний інститут" та інших вищих навчальних закладів, в яких розглядаються пріоритетні напрямки розвитку телекомунікаційних систем та мереж спеціального призначення.

ЗМІСТ

1. **Бандура А.В.** Моделювання ком'ютерних атак на основі *SQL*-ін'єкцій..... 4
2. **Васильців А.А.** Автоматизація пошуку потенційно-небезпечних дефектів реакції програм для впровадження кібернетичного впливу..... 5
3. **Волощук А.В.** Підсистема організації навчальної стратегії інтелектуальної комп'ютерної тренажерної системи навчання..... 6
4. **Головко Д.О., Сілко Д.О.** Модель розподілу прав доступу в інформаційних системах супроводу навчального процессу..... 7
5. **Горбенко В.І., Гіль А.В.** Програмні засоби моніторингу каналів передачі даних за протоколом *TCP/IP*..... 8
6. **Донець М.В.** Технологія програмування активних динамічних сполук об'єктів..... 9
7. **Думко Є.Ю., Сілко О.В.** Методика аналізу захищеності *web*-застосувань..... 10
8. **Жупанов О.П.** Формування раціональних варіантів перевірки об'єктів кібернетичного впливу..... 11
9. **Козак О.А.** Модель побудови системи управління мереж з комутацією пакетів..... 12
10. **Король М.А.** Аналіз засобів моніторингу інформаційних мереж та визначення ролі і задач системи підтримки прийняття рішень ідентифікації позаштатних ситуацій в інформаційній мережі..... 13
11. **Лазаренко М.О.** Управління засобами дослідження об'єктів комп'ютерної мережі..... 14
12. **Макарчук О.М., Кирган О.О.** Порівняльний аналіз криптографічних методів закриття інформації..... 15
13. **Нестеренко М.М., Висоцький Г.В.** Гібридні моделі забезпечення *QOS* в *IP*-мережах..... 16
14. **Рубановська Н.О.** Методика формування плану аналізу повідомлень оператором інформаційної мережі..... 17
15. **Сілко Д.О.** Модель мережевого елемента як об'єкта діагностування..... 18
16. **Терещенко М.М.** Організація прихованого каналу управління об'єктом інформаційного дослідження в технологіях мережевої розвідки..... 19
17. **Корнієнко А.Г., Зур А.А.** Розв'язання завдань дискретної оптимізації методом гілок та меж..... 20

Бандура А.В. (ВІТІ НТУУ "КПІ")

МОДЕЛЮВАННЯ КОМ'ЮТЕРНИХ АТАК НА ОСНОВІ *SQL*-ІН'ЄКЦІЙ

Додатки, засновані на *Web*-технологіях, знайшли широке застосування в різних галузях і стали невід'ємною частиною інформатизації обробки даних. Це призвело до ускладнення структури, архітектури і реалізації *Web*-додатків, що призвело до зниження стану їх захищеності.

Одним із найбільш небезпечних і поширених способів зламу *Web*-додатків і програм, що працюють з базами даних, робота яких заснована на *SQL* запитах, які виконуються системами керування баз даних (СКБД) є атаки типу *SQL*-ін'єкції. *SQL*-ін'єкції – це різновид атаки для отримання неавторизованого доступу до бази даних або для отримання інформації безпосередньо з бази даних. При недостатній фільтрації вхідних параметрів з додатку до СКБД, суб'єкт атаки може впроваджувати в запит довільний *SQL*-код. Суб'єкт може надсилати в об'єкт атаки параметри, які навмисно приведуть до помилки виконання *SQL*-запиту і змусять сервер видати повідомлення про помилку. По цих повідомленнях про помилку той, що атакує може зібрати необхідну інформацію для здійснення атак типу *SQL*-ін'єкції. Розробник може перехоплювати повідомлення про помилку, що генеруються сервером баз даних, але це не захищає додаток, а є лише іншою назвою методу *SQL*-ін'єкції.

Атаки типу *SQL*-ін'єкції, в залежності від типу СКБД, яка використовується, і умов впровадження, може дати можливість атакуючому виконати довільний запит до бази даних (наприклад, прочитати вміст будь-яких таблиць, видалити, змінити або додати дані), отримати можливість читання та / або запису локальних файлів та довільних команд на сервері, який атакується.

Для аналізу стану захищеності *Web*-додатку необхідна методика для побудови можливих шляхів його ураження. Підходом до вирішення цього завдання являється побудова моделей атак. Одним із варіантів моделювання комп'ютерних атак являється побудова дерева атак. Дерева атак являють собою опис варіантів проведення атаки для досягнення певної мети, яка ставиться на вершину дерева атаки. Кожен вузол в дереві являє собою деяку ціль, досягнення якої, в разі виконання низки умов, дозволяє зловмисникові піднятися по дереву на більш високий рівень і так до тих пір, поки зловмисник не досягне вершини дерева (кінцевої мети). В останньому випадку вважається, що зловмисник успішно реалізував атаку.

Варіанти атак, які можуть з'явитися тільки у винятково рідких випадках, можна взагалі опускати при побудові дерева атак, щоб не створювати занадто громіздких дерев. Варіанти, які можливі, але, швидше за все, з малою часткою ймовірності, слід позначати в дереві спеціальним чином. Застосування дерев атак дозволяє спростити завдання аналітиків при дослідженні проблеми безпеки та захищеності розроблюваного програмного продукту. Дерева атак володіють високою наочністю і дозволяють добре структурувати всілякі варіанти потенційних проблем для кожного з активів додатка. Для кожного активу додатка слід побудувати його дерево атак, в якому спробувати перелічити всі можливі шляхи здійснення атак на цей актив. Актив (або його властивість безпеки - конфіденційність, цілісність, доступність, автентичність і т.д.), є окремою частиною дерева.

Отже, для підвищення стану захищеності *Web*-додатків від атак типу *SQL*-ін'єкції, актуальними є застосування, які автоматично визначатимуть шляхи та методи впровадження цих атак. Для цього пропонується створення системи для автоматичої генерації дерев атак. Створивши дерево атак і проаналізувавши значення всіх його вузлів (ці значення будуть мінятися з часом, оскільки будуть уточнюватися відомості про можливі напади), можна використовувати цю схему для прийняття рішень з питань безпеки.

Васильців А.А. (ВІТІ НТУУ «КПІ»)

АВТОМАТИЗАЦІЯ ПОШУКУ ПОТЕНЦІЙНО-НЕБЕЗПЕЧНИХ ДЕФЕКТІВ РЕАКЦІЇ ПРОГРАМ ДЛЯ ВПРОВАДЖЕННЯ КІБЕРНЕТИЧНОГО ВПЛИВУ

На сьогоднішній час інформаційні технології впроваджуються у сфери діяльності людини. Не виключенням є військові організації, від рівня надійності яких напряму залежить національна безпека країни. Однак з стрімким розвитком виникає проблема захисту від комп'ютерних атак, техніка яких також постійно розвивається.

Для проведення кібернетичного впливу необхідна попередня інформаційна підготовка, одним із можливих елементів якої є пошук потенційно-небезпечних дефектів програм, які можна використати для проведення атаки.

На сьогодні існує цілий ряд засобів аналізу коду програм, як вихідного так і в двійковому вигляді, але всі вони дають результат у вигляді інформаційних повідомлень про можливу вразливість у тому чи іншому місці програми, інколи з рекомендаціями для усунення, частіше всього не враховуючи програмний засіб як цілісну систему взаємозалежних елементів виконання – що ускладнює пошук шляхів використання цих вразливостей в інтересах кібернетичного впливу, а це в свою чергу є не зовсім прийнятним результатом в умовах, коли від цього залежить хід політичних, економічних та інших міжнародних процесів в плані інтересів держави. Даним програмним засобам не вистачає математичного апарату, який би враховував не тільки особливості окремих елементів коду але й зробив би аналіз їх виконання на взаємодію в ході виконання програми.

Аналізатор вихідних текстів програми, який візуалізував взаємозв'язок окремих елементів програми з потенційно-небезпечними функціями – які призводять до порушення цілісності адресного простору процесу (зокрема, найбільш поширений тип вразливостей «переповнення буферу») – це допомогло б краще сприймати адресний простір коду та його вразливі ділянки коду, але не дозволяє враховувати можливий хід виконання коду – це дозволило би звузити варіанти пошуку ділянок, які можна використати для реалізації кібернетичного впливу.

Як відомо, математичний апарат теорії графів дозволяє проводити моделювання систем послідовного функціонування, що дає змогу в поєднанні з кінцевими автоматами накладати так звані контрольні точки на потенційно-небезпечні ділянки коду – при переході моделі програми в певну точку виконання, постійно слідкуючи за її елементами пам'яті можна знайти вразливу ділянку а також отримати повний звід щодо її використання – це досягається за допомогою попереднього створення шаблонів потенційно-небезпечних функцій, які потребують додаткового контролю при виконанні.

Для поєднання цих двох складових пошуку шляхів використання вразливостей, пропоную представляти модель коду програмного забезпечення у вигляді орієнтованого графа, вершинами якого будуть функції, реалізовані в коді, ребрами – переходи до інших функцій, при переході з однієї вершини на другу буде відбуватись порівняння стану змінних та і у випадку збігу з шаблоном – будуватиметься перелік рекомендацій щодо використання вразливої ділянки коду на основі взаєморозміщення та взаємовпливу елементів програми в адресному просторі під час її виконання – а після цього відбуватиметься обхід елементів всього графу з метою виявлення вразливостей та аналізу вимог для їх використанння.

Отже, враховуючи сучасні тенденції кібернетичного впливу, можна зробити висновок, що дана пропозиція є перспективною та потрібною для впровадження з метою забезпечення кібернетичної стійкості в сфері інформаційної боротьби а також для впровадження в навчальний процес при підготовці фахівців в даній сфері.

Волощук А.В. (ВІТІ НТУУ «КПІ»)

ПІДСИСТЕМА ОРГАНІЗАЦІЇ НАВЧАЛЬНОЇ СТРАТЕГІЇ ІНТЕЛЕКТУАЛЬНОЇ КОМП'ЮТЕРНОЇ ТРЕНАЖЕРНОЇ СИСТЕМИ НАВЧАННЯ

На сьогоднішній день підготовка висококваліфікованих кадрів є актуальною задачею для будь-якої організації або установи. Сучасний навчальний процес складно уявити без використання комп'ютерних підручників, тренажерів, лабораторних практикумів, тестуючих, контролюючих та інших комп'ютерних систем навчання. Одним з перспективних напрямків розвитку комп'ютерних систем навчання є розробка інтелектуальних комп'ютерних тренажерних систем навчання (ІКТСН), що для свого проектування використовують методи та засоби штучного інтелекту (ШІ).

ІКТСН складається з трьох підсистем – підсистеми конструювання навчально-тренажних задач, тренажно-імітаційної підсистеми та підсистеми інтерфейсу. Кожна з підсистем тісно взаємодіє одна з одною, використовуючи для цього модуль організації і управління навчальною стратегією.

Для організації роботи підсистеми конструювання навчально-тренажних задач та тренажно-імітаційної підсистеми використана продукційна модель представлення знань. Вона отримала найбільше поширення серед інших моделей представлення знань.

Продукційна модель або модель, заснована на правилах, дозволяє представити знання у вигляді виразів типу „ЯКЩО (умова), ТОДІ (дія)".

Під „умовою" (антецедентом) розуміється деякий вираз-зразок, по якому відбувається пошук в базі знань, а під „дією" (консеквентом) – дію, виконувану при успішному результаті пошуку (вони можуть бути проміжними, виступаючими далі як умови, і термінальними або цільовими, які завершують роботу системи.

При використанні продукційної моделі база знань складається із набору правил. Програма, яка керує перебором правил, називається машиною виводу. В ІКТСН її роль виконує тренажно-імітаційна підсистема.

Машина виводу (інтерпретатор правил) виконує дві функції: по-перше, перегляд існуючих фактів із робочої пам'яті (бази даних) і правил із бази знань і додавання (по мірі можливості) в робочу пам'ять нових фактів, і по-друге, визначення порядку перегляду і застосування правил. Цей механізм керує процесом консультації, зберігаючи для користувача інформацію про отримані висновки, і запитує у нього інформацію, коли для спрацювання чергового правила в робочій пам'яті знаходиться недостатньо даних.

Інтерпретатор продукцій працює циклічно. В кожному циклі він переглядає всі правила, щоб виявити те, умова якого співпадає з відомими на даний момент фактами із робочої пам'яті. Після вибору правило спрацьовує, його висновок заноситься в робочу пам'ять, і потім цикл повторюється знову.

Для забезпечення максимальної інформативності і об'єктивності результатів контролю було запропоновано впровадити в логіку роботи модулю конструювання програмної логіки навчально-тренувальної задачі ІКТСН адаптивні алгоритми.

Запропонований алгоритм оцінки рівня засвоєння навчального матеріалу за результатами практичної роботи в інтелектуальній комп'ютерній тренажерній системі навчання підвищує якість підготовки спеціалістів з інформаційних технологій за рахунок інтеграції адаптивних моделей теорії *IRT*, що визначає наукову новизну роботи.

Відмінністю запропонованого рішення від існуючих є застосування методів параметричної моделі *IRT* для визначення ступеня засвоєння навчального матеріалу та методу збалансованості М. Челишкової для вирішення завдання калібровки коефіцієнтів тестових завдань.

Головко Д.О. (ВІТІ НТУУ „КПІ”)
Сілко Д.О. (ВІТІ НТУУ „КПІ”)

МОДЕЛЬ РОЗПОДІЛУ ПРАВ ДОСТУПУ В ІНФОРМАЦІЙНИХ СИСТЕМАХ СУПРОВОДУ НАВЧАЛЬНОГО ПРОЦЕСУ

Основною проблемою інформаційних систем супроводу навчального процесу (ІС СНП) є розподіл прав доступу (*access rights*). Права доступу до інформації – сукупність правил, що регламентують порядок і умови доступу суб'єкта до інформації, встановлених правовими документами чи власниками інформації. Права доступу визначають набір прав (читання, запис, виконання), дозволених для виконання суб'єктам (наприклад, користувачам системи) над об'єктами даних.

Аналіз сучасних ІС СНП показав, що їх більшість контролюють права доступу на рівні типів об'єктів системи, або безпосередньо через механізм надання прав для пари користувач-запис.Розробники ІС СНП безпосередньо в коді програмиреалізують механізм надання динамічних прав, тобто доступна множинаоб'єктів вказується не вручну, а описується допомогою визначеної логічної умови.

Найбільш перспективним підходом до вирішення питання розподілу прав доступу є застосування методу, який заснований на ролях. Суть підходу полягає в динамічній генерації *SQL* запитів, при чому опис правил генерації здійснюється розробниками спеціалізованої конфігурації, а вхідні параметри для процедури генерації вказують адміністратори системи.

При прийнятті рішення про надання доступу звичайно аналізується наступна інформація:

– ідентифікатор суб'єкта (ідентифікатор користувача, мережеву адресу комп'ютера);

– атрибути суб'єкта (позначка безпеки, група користувача).

Для кожного типу об'єкта існує свій набір прав, тому можливо виділити такі:

1. Права, які на виході для кожного користувача повинні повернути доступну для нього множину об'єктів.

2. Права, які для пари користувач-об'єкт повинні повернути булеве значення: чи має право на виконання операції чи ні.

Другий тип прав можливо звести до першого шляхом використання процедури приналежності цього об'єкта до дозволенного для користувача множини об'єктів.

Нехай A_i –множина об'єктів типу i, γ_{ijk} – оператор доступних об'єктів типу i права j для суб'єкта k.Для визначення права типу 1 і використовується данамножина $A_{ijk} = \gamma_{ijk}(A_i)$. Для визначення права типу 2 перевіряємо умову: $\gamma \in \gamma_{ijk}$, де $\gamma \in A_i$. Якщо ця умова виконується, то суб'єкт k має право виконати операцію j над цим об'єктом.

Введення класів і підкласів доступу суб'єктів до об'єктів контролю доступу забезпечує можливість гнучкої та динамічної зміни прав доступу суб'єктів до об'єктів при корегуванні політики безпеки в ІС СНП.

На основі наведених міркувань описаний підхід щодо реалізації моделі розподілу прав доступу до об'єктів в ІС СНП. Дана модель значно поліпшить захист інформації від несанкціонованого доступу, та корегування інформації в ІС.

ЛІТЕРАТУРА

1. *Петров В.А.*, *Піскарьов С.А.*, Шеїн А.В. Захист інформації від несанкціонованого доступу в автоматизованих системах. – М.: Інформаційна безпека, 2001. – С.21-35.

2. *Карасик І.* Програмні та апаратні засоби захисту інформації для персональних комп'ютерів // Компьютер Пресс № 3, 1995.

Донець М.В. (ВІТІ НТУУ «КПІ»)

ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ АКТИВНИХ ДИНАМІЧНИХ СПОЛУК ОБ'ЄКТІВ

Технологія об'єктно-орієнтованого програмування (ООП) на сьогодні є домінуючою, вона реалізована практично в усіх сучасних мовах програмування. Створено багато інструментальних засобів автоматизації проектування програмного забезпечення (ПЗ) на основі ООП, які дозволяють побудувати об'єктну модель предметної області.

Але ООП не дає відповіді на запитання, як все ж таки з об'єктів побудувати програму. Великий обсяг допоміжного програмного коду, який забезпечує функціонування об'єктів, з'єднує об'єкти в єдину програму та примушує їх взаємодіяти, залишається поза межами об'єктно-орієнтованого проектування. Тому розробка технології програмування, яка б забезпечувала автоматизацію створення закінченого ПЗ, є актуальною проблемою.

Одним з підходів до створення ПЗ є побудова програми у вигляді активних динамічних сполук об'єктів, надалі його будемо називати АДС-технологією програмування.

Концепція АДС-технології базується на таких положеннях:

–об'єкт є програмним модулем, який перетворює вхідні значення в вихідні значення;

–об'єкт може отримувати управління для виконання перетворення вхідних значень в вихідні значення;

–процес від початку зміни вхідних значень до закінчення обчислень всіх вихідних значень може виконуватись транзакційно, з відкатом змін або без відкату;

–об'єкт може пасивно взаємодіяти з іншими об'єктами будь-яких класів шляхом читання значень їх властивостей;

–об'єкт може активно взаємодіяти з іншими об'єктами будь-яких класів шляхом передавання їм управління (активації).

Можна сформулювати таке бачення процесу програмування з використанням АДС-технології:

– об'єкти можуть створювати динамічні сполучення шляхом з'єднання вихідних властивостей одних об'єктів з вхідними властивостями інших об'єктів, а також отримувати можливість активації сполученого об'єкта;

– будь-яка програма може бути побудована у вигляді сполук об'єктів;

– процес розробки програми можна звести до розробки класів та синтезу сполук об'єктів; цей процес можна візуалізувати в інтегральному середовищі розробки;

– один об'єкт в різні моменти часу виконання програми може брати участь в різних сполуках, отже, з одних й тих самих об'єктів можна динамічно під час виконання програми синтезувати різні сполуки;

– сполуки об'єктів можна оформляти у вигляді класів сполук, які можуть з'єднуватися з іншими сполуками, утворюючи складні поліморфні конструкції;

– можливе створення інтерактивних програм, в яких користувачі самі синтезуватимуть потрібні сполуки для розв'язування своїх задач або використовуватимуть інтелектуальний інтерфейс для автоматичного синтезу потрібної сполуки за заданою постановкою задачі.

Перспективними напрямами досліджень з розвитку АДС-технології є:

– розробка теоретичних основ проектування схем сполук об'єктів;

– створення мови опису схем сполук об'єктів для їх динамічного синтезу;

– розробка методу синтезу потрібної схеми сполуки за заданою постановкою задачі;

– дослідження шляхів створення програм з можливістю адаптації до умов експлуатації.

Думко Є.Ю.(ВІТІ НТУУ «КПІ»)
Сілко О.В.(ВІТІ НТУУ «КПІ»)

МЕТОДИКА АНАЛІЗУ ЗАХИЩЕНОСТІ *WEB*-ЗАСТОСУВАНЬ

Помилки в захисті веб-додатків, сайтів та блогів на сьогоднішній день залишаються одним з найбільш поширених недоліків забезпечення захисту інформації. Більше того, вразливості веб-додатків є одним з найбільш поширених шляхів реалізації зловмисниками атак на *web*-застосування з метою крадіжки інформації та наступним проникненням в інформаційні системи. Зусилля розробників веб-додатків зазвичай спрямовані головним чином на вирішення завдань, пов'язаних з реалізацією функцій системи. Питанням безпеки та якості програмного коду приділяється недостатньо уваги. В результаті переважна більшість веб-додатків містить вразливості різного ступеня критичності.

Проектування безпечних веб-застосувань, як правило, засноване на декількох ключових принципах: всі дані, які введені до веб-додатку, вважаються зловмисними; область веб-додатку, доступна для атак, повинна бути мінімізована; доступ до веб-застосування за замовчуванням блокується; використовується ешелонована оборона.

Для розробки веб-додатків застосовуються різноманітні інструментальні засоби та мови програмування. Зокрема широко поширені динамічні мови програмування, такі як *PHP*, *Perl*, *Python*.

Застосування методу статичного аналізу захищеності до *web*-застосувань, які побудовані на основі динамічних мов має істотний недолік: у додатках, де вихідний код генерується під час виконання, метод статичного аналізу не може гарантувати виявлення всіх вразливостей, пов'язаних з порушенням властивості *non interference*. Метод динамічного аналізу захищеності також не може гарантувати виявлення всіх вразливостей, пов'язаних з порушенням властивості *non interference*, якщо тестові набори даних не покривають всі можливі шляхи виконання програми. Методи статичного і динамічного аналізу дозволяють виявляти тільки вразливості, пов'язані з передачею веб-додатку некоректних вхідних даних. Метод тестування на проникнення також не може гарантувати виявлення всіх вразливостей. Таким чином, всі зазначені методи виявлення вразливостей веб-додатків не можуть гарантувати виявлення всіх видів вразоивостей *web*-застосувань. З проведеного аналізу можна зробити висновок, що подальший розвиток методів автоматичного виявлення вразливостей веб-додатків повинен будуватися на інтеграції можливостей різних методів, що дозволить:

- охопити максимально широкий клас вразливостей;
- контролювати повноту виявлення вразливостей, щоб по підсумкам автоматичного аналізу можна було б (в ідеалі) дати гарантію відсутності вразливостей заданих класів.

ЛІТЕРАТУРА:

1. *Reto E.Heney, Lance J.Hoffman. Information warfare //Institute of Engineering and Applied Sciences. At George Washington University. - Washington, December 1995.* Режим доступу до журналу: *http://www.seas.gwu.edu/student/reto/.*
2. *D. Wagner and R. Dean. Intrusion detection via static analysis. In Proc. of the 2001 IEEE Symposium on Security and Privacy, pages 156-169, Los Alamitos, CA, May 14-16 2001.*
3. *J. Allen et al. State of the practice of intrusion detection technologies. TR CMU/SEI-99-TR-028, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Jan. 2000.*
4. *Steng. D, Mun S. Secrets of network security. K.: Dialectics, 1995. 544 p.*
5. *Classifying information for security. // In : DataPro on information security, 1989, IS15-250-101-106.*

Жупанов О.П. (ВІТІ НТУУ «КПІ»)

ФОРМУВАННЯ РАЦІОНАЛЬНИХ ВАРІАНТІВ ПЕРЕВІРКИ ОБ'ЄКТІВ КІБЕРНЕТИЧНОГО ВПЛИВУ

Для забезпечення незалежності і суверенітету держави в умовах сьогодення збройним формуванням недостатньо мати лише натреновану ну бойові дії армію, вогневі засоби та техніку. Останнім часом у світі набуває популярності інший спосіб вирішення конфліктів – кібернетичний вплив. Для ефективної реалізації кібернетичного впливу суб'єкт атаки повинен мати відомості для визначення цілей програми атаки. Отримання таких відомостей є результатом проведення інформаційної підготовки суб'єктом атаки.

Інформаційна підготовка проводиться інженером що безпосередньо відповідає за проведення кібернетичної атаки. Основним способом проведення інформаційної підготовки атаки є аналіз реакції об'єктів комп'ютерної мережі на запити що посилає об'єкт проведення підготовки. Аналіз проводиться на основі даних про час реакції об'єкту атаки на отриманий запит. Оскільки дані про початок атаки можуть бути недоступними в момент підготовки атаки з'являється поняття адекватності даних що будуть використані в момент атаки. Ключовим питанням організації інформаційної підготовки є розробка раціональних способів перевірки потенційних об'єктів атаки. Таким чином задача інформаційної підготовки до масового кібернетичного впливу аналогічна загальній математичній задачі про комівояжера". Сутність задачі полягає у знаходженні оптимального порядку перевірки ("обходу") певної множини об'єктів будь-якої природи, щоб цільова функція накладних витрат на таку роботу була б мінімальною. Таким чином задача комівояжера при відображенні на процес інформаційної підготовки до кібернетичного впливу полягає у знаходженні раціонального способу перевірки об'єктів комп'ютерної мережі.

Задача організації збору даних не є суттєвою при невеликій кількості об'єктів мережі, але на даний момент глобальна мережа щодня поповнюється мільйонами користувачів по всьому світі. Навіть сучасні комп'ютери потужністю 5-10 трильйонів операцій за секунду не підходять для вирішення такої задачі. Тому за мету роботи поставлено розробку системи що б використовувала дієвий алгоритм що б забезпечив як адекватність та наявність необхідних даних так і не перевищував існуючі обмеження до обчислювальних ресурсів.

У якості такого алгоритму був використав «мурашиний алгоритм» як оптимальний для розв'язання задач на графах. Так, саме на графах, адже усю комп'ютерну мережу що полягає обробці та спостереженню зручно представити у вигляді графу, в якому в якості вузлів будуть виступати об'єкти комп'ютерної системи (комп'ютери, обчислювальні пристрої різного призначення), а в якості зв'язків між вузлами буде виступати можливість взаємного обміну даними між цими вузлами. Мурашиний алгоритм − один з ефективних поліноміальних алгоритмів для знаходження наближених рішень задачі комівояжера, а також аналогічних завдань пошуку маршрутів на графах. Суть підходу полягає в аналізі та використанні моделі поведінки мурах, що шукають дороги від колонії до їжі. В природі, мурахи (початково) блукають довільним чином, і по знаходженні їжі повертаються до колонії, залишаючи по собі феромоновий слід. В даному випадку феромон це технологія обміну даними між мурахами. Якщо інші мурахи знаходять такий шлях, вони схильні припинити свої блукання, натомість слідувати позначеним шляхом, посилюючи його під час повернення у разі знайдення їжі.

Оцінка ефективності розробленої системи показує що метод повного перебору усіх доступних маршрутів доцільний до використання при кількості вузлів що не перевищує 60. З аналогічними параметрами системи мурашиний алгоритм може обробляти графи розміром до 500 вузлів з необмеженою кількістю зв'язків між вузлами за прийнятні проміжки часу (5-10 сек). Єдиним недоліком таких алгоритмів є наближений розв'язок.

Козак О.А. (ВІТІ НТУУ „КПІ”)

МОДЕЛЬ ПОБУДОВИ СИСТЕМИ УПРАВЛІННЯ МЕРЕЖ З КОМУТАЦІЄЮ ПАКЕТІВ

Сучасний розвиток інформаційно-телекомунікаційних мереж характеризується наданням все більшого спектру послуг користувачам, постійним збільшення об'єму різнорідного трафіку та ускладненням мережевих технологій, що вимагає створення ефективної системи управління мережею. Це пояснюється тим, що оптимальність структури та конфігурування системи управління є визначним фактором ефективності роботи транспортної мережі щодо забезпечення необхідного рівня якості обслуговування в аварійних ситуаціях та перевантаженні.

Основним завданням системи управління довгий час було зменшення часу реакції експлуатаційних служб на зміну стану мережі. Існуючі сучасні мережеві технології дозволяють говорити про інтегральне управління, тому вони застосовують мережеве обладнання з програмним управлінням, дистанційно-керованими системами комутації на всіх рівнях ієрархії.

На даний час, існують рекомендації *ITU-T X*.700 та близький до них стандарт *ISO* 7498-4, які ділять задачі системи управління на п'ять функціональних груп:

–- управління конфігурацією мережі та ідентифікації її елементів;
–- обробка помилок;
–- аналіз продуктивності і надійності;
–- управління безпекою;
–- облік роботи мережі.

В свою чергу в склад базової архітектури управління мережами входять наступні основні елементи:

1. Система управління мережею (СУМ).
2. Об'єкти управління.
3. Інформаційна база управління.
4. Протокол управління мережею.

Також необхідно відмітити, що від вибір типу архітектури системи управління має величезний вплив на її продуктивність. Існує декілька типових архітектур побудови системуправління мережею:

–однорівнева архітектура;
–ієрархічна архітектура;
–комірчаста архітектура;
–платформна архітектура.

Представлена модель системи управління локальною мережею *SOWA* відноситься до систем управління централізованого типу. В свою чергу це розподілена система управління що в комплексі дозволяє, з одного боку, зосередити всі управляючі ресурси в одному місці, а з іншого боку, забезпечить можливість здійснення доступу до цих ресурсів з різних точок мережі.

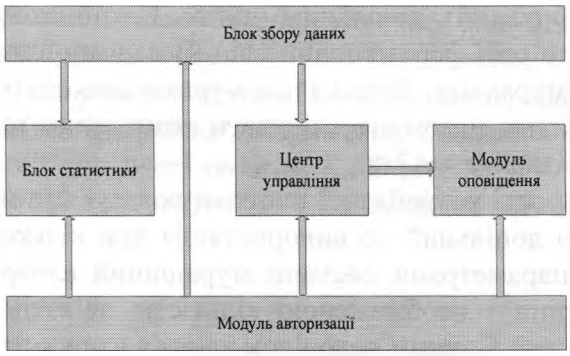

Рис. 1. Структура СУМ *SOWA*

Король М.А. (ВІТІ НУТУ "КПІ")

АНАЛІЗ ЗАСОБІВ МОНІТОРИНГУ ІНФОРМАЦІЙНИХ МЕРЕЖ ТА ВИЗНАЧЕННЯ РОЛІ І ЗАДАЧ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ІДЕНТИФІКАЦІЇ ПОЗАШТАТНИХ СИТУАЦІЙ В ІНФОРМАЦІЙНІЙ МЕРЕЖІ

На поточному етапі розвитку суспільства в галузі інформаційних технологій (ІТ) спостерігається стрімке розширення спектру послуг та інтеграція телекомунікаційних мереж різноманітного призначення. Для забезпечення високої якості обслуговування користувачів під час впровадження сучасних ІТ і забезпечення високотехнологічних інформаційних послуг потрібна відповідна система управління та моніторингу інформаційної мережі, у складі якої має функціонувати підсистема підтримки прийняття рішень черговим (оперативним) персоналом. Зазначена система має надавати персоналу можливість проведення моніторингу і швидкого реконфігурування мережі, своєчасно знаходити несправності та заздалегідь повідомляти про їх можливе виникнення.

Розробка і впровадження засобів організаційно-технічного та програмного забезпечення діяльності персоналу є одним з важливих питань функціонування інформаційних систем структурних підрозділів ЗСУ. На персонал покладається відповідальність за усунення невизначеності та прийняття рішень щодо оцінки негативного впливу на поточний стан інформаційної системи.

Для ефективної діяльності такого персоналу потрібна безперервна робота системи моніторингу. У зв'язку з цим накопичується дуже великий файл з інформацією про діяльність самої системи та її користувачів, який в майбутньому і в найкоротший термін потрібно обробити. З досвіду можна сказати, що об'єм інформації, який потрібно обробити адміністратору, значно перевищує оптимальний об'єм, який він в змозі обробити. З цього випливає, що час реакції оператора системи на певну проблему в мережі не дає змогу оперативно реагувати на події, що відбуваються в мережі. Тобто потрібно автоматизувати рутинні операції обробки великої кількості подій щодо реагування на невідомі досі події.

Методи аналізу інформації, що використовуються в сучасних системах моніторингу даних в інформаційно-аналітичних системах є достатньо ефективними якщо відомі точні характеристики подій. У ході дослідження було виявлено, що системи моніторингу, наприклад, *HPOpenView*, *SunNetManager*, *IBMNetView*, *MRTG*, *Nagios*, збирають значну кількість інформації, але не виконують навіть найпростішого аналізу.

Всі вони передбачають застосування методів, що пов'язані зі збором інформації про стан вузлів інформаційно-аналітичної системи. Але вручну переглядати всі нестандартні ситуації, що виникають, є досить об'ємною задачею, тому всю отриману для аналізу інформації потрібно систематизувати.

Переважна більшість сучасних системи моніторингу даних інформаційної мережі зустрічаються з однією і тією ж проблемою: характеристики проблем, що виникають, потребують швидкої і оптимально-правильної реакції системи, яка була б спроможна залишатись ефективною, навіть при невідомих точних характеристиках несправності.

Отже, для ефективної роботи системи моніторингу функціонування вузлів в інформаційній мережі потрібно автоматизувати роботу персоналу, який приймає рішення, а також використовувати методи, що дозволяли б реагувати на нові нестандартні ситуації в роботі інформаційної системи. Одним із способів вирішення даної задачі є використання системи підтримки прийняття рішень в інформаційній мережі моніторингу функціонування її вузлів. Використання інтелектуальних технологій дасть змогу фахівцям ІТ аналізувати процеси функціонування комп'ютерних систем, виявляти приховані залежності у даних, отримувати моделі, що дозволяють оцінювати та прогнозувати виникнення можливих проблем з метою їх запобігання і, таким чином, підвищити рівень працездатності комп'ютерної системи в цілому.

Лазаренко М.О. (ВІТІ НТУУ «КПІ»)

УПРАВЛІННЯ ЗАСОБАМИ ДОСЛІДЖЕННЯ ОБ'ЄКТІВ КОМП'ЮТЕРНОЇ МЕРЕЖІ

Аналіз моделей мережевого оточення об'єкту кібернетичного впливу в комп'ютеризованих системах є одним з базових питань розробки програмно-технічних методів силового кібернетичного впливу. Найбільш придатним апаратом для вирішення задач даного типу вважаються розподілені системи сканування мережевого оточення.

Важливим аспектом аналізу мережевого оточення об'єкту кібернетичного впливу як складної мережі полягає у відображенні прагматично важливих властивостей об'єкту кібернетичного впливу.

При обробці великої кількості вузлів (від 1000-100000) тривалість сканування проходить за великий проміжок часу. При проведенні інформаційної підготовки з одного вузла ми не можемо домогтися високої швидкодії через обмежену пропускну здатність мережі. Метою наукової роботи ставилось зменшити часові показники при проведенні інформаційної підготовки масованого кібернетичного впливу при наявності апріорних часових показників. Для скорочення часу сканування вузлів мережі пропонується зробити розподілену систему типу «Менеджер - Агент». Розподілена система - це набір незалежних комп'ютерів, що представляється їх користувачам єдиної об'єднаної системою. Розподілені системи складаються з автономних комп'ютерів, які працюють спільно, постаючи у вигляді єдиної зв'язної системи. Існує шість основних характеристик розподілених систем за допомогою яких досягається мета наукової роботи:

1. Спільне використання ресурсів. Розподілені системи допускають спільне використання як апаратних (жорстких дисків, принтерів), так і програмних (файлів, компіляторів) ресурсів.
2. Відкритість. Це можливість розширення системи шляхом додавання нових ресурсів до вже існуючих.
3. Паралельність. У розподілених системах кілька процесів можуть одночасно виконуватися на різних комп'ютерах в мережі. Ці процеси можуть взаємодіяти під час їх виконання та синхронізувати свою роботу.
4. Масштабованість. Під масштабованістю розуміється можливість додавання у систему нових властивостей і методів.
5. Відмовостійкість. Наявність декількох комп'ютерів дозволяє дублювання інформації і стійкість до деяких апаратним і програмним помилок. Розподілені системи в разі помилки можуть підтримувати часткову функціональність. Повний збій в роботі системи відбувається тільки при мережевих помилки.
6. Прозорість. Користувачам надається доступ до ресурсів в системі, в той же час від них прихована інформація про розподіл ресурсів по системі.

В ході дослідження інформаційної підготовки було запропоновано створити програмно-технічний комплекс використовуючи сучасні технології розподілених систем який би зменшував витрати часових показників при аналізі великої кількості вузлів мережі.

Отже зменшення часових показників при проведенні інформаційної підготовки масованого кібернетичного впливу пропонується шляхом використовувати розподілені системи управління типу «Менеджер - Агент», що буде зменшувати часові показники при збільшенні його кількісного показника.

Доповідь присвячена висвітленню варіанта використання розподілених систем при вдосконаленні проведення інформаційної підготовки як важливої складової масовоного кібернетичного впливу на елементи комп'ютеризованої системи.

к.т.н. Макарчук О.М. (ВІТІ НТУУ „КПІ”)
Кирган О.О. (ВІТІ НТУУ „КПІ”)

ПОРІВНЯЛЬНИЙ АНАЛІЗ КРИПТОГРАФІЧНИХ МЕТОДІВ ЗАКРИТТЯ ІНФОРМАЦІЇ

На сьогодні в інформаційному просторі, швидкими темпами впроваджуються новітні досягнення комп'ютерних і телекомунікаційних технологій. Комп'ютерні системи активно впроваджуються у фінансові, промислові, торгові і соціальні сфери. Внаслідок цього різко зріс інтерес широкого кола користувачів до проблем захисту інформації. Захист інформації – це сукупність організаційно-технічних заходів і правових норм для попередження заподіяння збитку інтересам власника інформації.

Розглядаються класичні та сучасні підходи до побудови та аналізу криптографічних алгоритмів та криптосистем. Значна увага звертається на важливість теоретичного аналізу коректності та надійності криптографічних алгоритмів. Вводяться поняття криптографії та криптоаналізу, надійності та ефективності криптосистем.

Наведено формальне визначення криптосистеми, властивості шифрувальних відображень, шифри, що утворюють групу. Розглянуто деякі математичні аспекти (класичний та розширений алгоритми Евкліда, групи та кільця по модулю, арифметика лишків, конгруенції), які використовуються в процесі побудови та аналізу криптографічних алгоритмів.

Серед всього спектру методів захисту даних від небажаного доступу особливе місце займають криптографічні методи. Криптографія – наука про математичні методи забезпечення конфіденційності і автентичності інформації.

З метою визначення найдоцільнішого криптографічного алгоритму при проектуванні захищеної системи, авторами статті зроблено порівняльний аналіз алгоритмів шифрування з огляду на наступні критерії: високий рівень захисту даних проти дешифрування та можливої модифікації; захищеність інформації повинна базуватися тільки на знанні ключа та не залежати від того, відомий алгоритм чи ні; невеликі зміни вихідного тексту або ключа має приводити до значних змін шифрованого тексту; область значень ключа має виключати можливість дешифрування даних шляхом перебору значень ключа; економічність реалізації алгоритму при достатній швидкодії; вартість дешифрування даних без відомого ключа має перевищувати вартість самих даних.

Сучасні алгоритми шифрування передбачають, що стороннім особам може бути відомий як сам алгоритм, так і частини розшифрованого відкритого коду. Єдине, що не знає “викрадач інформації” це секретний ключ, що не дозволяє розшифрувати дані. Єдиним методом відкриття секретного ключа є послідовний перебір всіх можливих варіантів. Вважається, що довжина секретного коду, що використовуються в сучасних алгоритмах шифрування даних не дозволяє розкодувати дані за прийнятний час. До алгоритмів шифрування також висуваються вимоги відносної стійкості криптографічному аналізу.

Отже, в цій роботі було зроблено огляд найбільш поширених в даний час методів криптографічного захисту інформації та способів її реалізації. Вибір для конкретних систем повинен бути заснований на глибокому аналізі слабких і сильних сторін тих або інших методів захисту. Обгрунтований вибір тієї чи іншої системи захисту взагалі ж повинен спиратися на якісь критерії ефективності. На жаль, до цих пір не розроблені відповідні методики оцінки ефективності криптографічних систем. Найбільш простий критерій такої ефективності – імовірність розкриття ключа або потужність множини ключів (М). По суті це те ж саме, що і крипостійкість. Для її чисельної оцінки можна використовувати також і складність розкриття шифру шляхом перебору всіх ключів.

к.т.н. Нестеренко М.М. (ВІТІ НТУУ „КПІ”)
Висоцький Г.В. (ВІТІ НТУУ „КПІ”)

ГІБРИДНІ МОДЕЛІ ЗАБЕЗПЕЧЕННЯ *QOS* В *IP*-МЕРЕЖАХ

Для сучасних інформаційно-телекомунікаційних мереж характерним є необхідність передачі повідомлень різного типу, при цьому існує тенденція збільшення об’ємів передачі мультимедійного трафіка. Одним з основних завдань управління такою мережею є організація ефективної доставки інформації. Тому, забезпечення якості обслуговування (*Quality of Service – QoS*), для всіх видів трафіку, є досить актуальною задачею.

Для забезпечення виконання показників *QoS* в *IP*-мережах використовуються наступні методи: резервування ресурсів (на час з’єднання запитуються і резервуються всі необхідні, для виконання програмних додатків, ресурси); пріоритизація трафіку (поділ трафіку в мережі на класи з пріоритетним порядком обслуговування деяких з них); перемаршрутізація (дозволяє при перевантаженні в мережі перевести трафік на резервний маршрут; саме цим способом забезпечується *QoS* у переважній більшості контролерів *Session Border Controller –SBC*).

У сучасних *IP*-мережах перераховані методи найшли використання в: моделі інтегрованого сервісу – *Integrated Service* (*IntServ*); моделі диференційних послуг – *Differentiated Service* (*DiffServ*); гібридній моделі для надання *QoS* з кінця в кінець *IntServ-DiffServ.* Основною перевагою *IntServ* являється забезпечення гарантованої *QoS,* шляхом класифікації кожного інформаційного потоку даних та відповідно: резервуванням необхідної смуги пропускання в каналі зв’язку, забезпечення мінімальної затримки при передачі пакетів або мінімальний рівень їх втрат в залежності від класу обслуговування.

Найбільш істотний недолік *IntServ* пов’язаний з масштабованістю *Resource ReSerVation Protocol* (*RSVP*), який є основою даної моделі. Особливо це проблематично для високошвидкісних магістральних мереж, оскільки *RSVP* проводить резервування тільки для одного інформаційного потоку та для підтримання актуальності кожного з’єднання необхідно використовувати достатньо великий обсяг сигнальної інформації.

В зв’язку з цим, була запропонована модель *DiffServ*, особливість якої полягає в забезпеченні параметрів *QoS* на підставі вимог різних груп користувачів, диференціюючи трафік за встановленим класом (пріоритетом). Тобто, вхідний трафік ідентифікується, нормалізується, профілюється та реалізується пріоритетне обслуговування в активному обладнанні мережі.

Недолік даної моделі полягає в тому, що в випадку перевантажень в мережі затримки передачі можуть мати досить велике значення, а пакети з меншим пріоритетом будіть просто відкинуті.

Найбільш перспективною на сьогодні є модель *Int-DiffServ* завдяки тому, що використовує кращі сторони представлених моделей на різних ділянках мережі. Домінуючим компонентом *Int-DiffServ* є технологія мультипротокольної комутації по мітках (*Multiprotocol Label Switching – MPLS*), яка призначена для зменшення часу передачі *IP*-пакетів за рахунок комутації останніх на основі добавлених міток. Однак, *MPLS* стає технологією з контролем забезпечення *QoS* тільки при використанні *RSVP-TE.*

В свою чергу, *RSVP-TE* не лише резервує необхідний рівень, якості обслуговування на всіх ділянках встановленого маршруту, але й має функцію динамічного переведення на резервний маршрут в випадку перевантажень або виходу з ладу елементів мережі.

Однак, при використанні запропонованих мережевих технологій виникає ряд питань стосовно їх переваг в порівнянні з традиційною *IP*-маршрутизацією. Тому, для вирішення задачі оцінки величини ефекту застосування *MPLS* необхідно розробити адекватні математичні моделі, які будуть визначати апаратні та програмні витрати активного обладнання в залежності від розмірності мережі та протоколів передачі та сигналізації.

Рубановська Н.О. (ВІТІ НТУУ "КПІ")

МЕТОДИКА ФОРМУВАННЯ ПЛАНУ АНАЛІЗУ ПОВІДОМЛЕНЬ ОПЕРАТОРОМ ІНФОРМАЦІЙНОЇ МЕРЕЖІ

Розробка методики плану аналізу повідомлень оператором інформаційної мережі зараз є актуальним питанням, але в цьому напрямку зроблені лише перші кроки. В Україні помітне суттєве відставання як у теоретичному так і у практичному плані, зокрема в питаннях адміністрування інформаційних мереж (ІМ), де дуже важливо прийняття обґрунтованих і своєчасних рішень.

Інформація щодо технічного стану інформаційних напрямків ІМ та функціонування підсистем потрапляє до оператора ІМ шляхом застосування спеціального програмного забезпечення через дискретні відмітки часу. Повна оцінка інформації можлива тільки після послідовного ознайомлення з усіма елементами вихідної інформації за всіма інформаційними напрямками, що в значній мірі погіршує нормативні часові показники щодо забезпечення стійкого функціонування ІМ і відновлення її основних показників у разі виникнення позаштатних ситуацій

Таким чином, актуальною стає задача обробки усієї вхідної інформації щодо стану функціонування ІМ в реальному часі з можливістю оперативного втручання в управління мережею її оператором.

Виявлення та усунення нештатних ситуацій є однією з головних функцій оператора ІМ. В інструкціях оператора є рекомендації по діях в найбільш типових нештатних ситуаціях. Разом з цим, якщо ситуацію не можливо віднести ні до одного з наведених типів, то заходи щодо її усунення та рішення щодо подальшого функціонування ІМ приймаються пізніше після повної обробки і аналізу інформації групою експертів (фахівців).

Досвід експлуатації ІМ показує, що вірогідність розпізнавання нештатної ситуації та вироблення обґрунтованого рішення щодо подальшого функціонування ІМ черговим адміністратором на протязі однієї години дуже мала.

Формування плану аналізу оператором повідомлень про несправності може бути здійснено шляхом вирішення задачі ранжування несправностей у порядку зменшення їх важливості та терміновості.

Ця задача інтерпретується як задача складання розкладу. Кожна несправність $S^{(i)}$ характеризується наступними величинами: i – індекс несправності; T_i^d – директивний час аналізу повідомлення про i-у несправність; t_i – час аналізу повідомлення про i-у несправність; w_i – коефіцієнт відносної важливості i-ї несправності.

Відомо, що задачі складання оптимального розкладу не піддаються рішенню аналітичними методами та є *NP*-повними. Для їхнього вирішення застосовуються точні та наближені методи. Точні методи у наслідок їхньої великої обчислювальної складності не прийнятні для реалізації у режимі часу, близькому до реального. Тому, для вирішення задачі формування плану обробки повідомлень про несправності оператором ІМ, цілком доцільно застосувати один з найбільш ефективних наближених методів.

Найбільш відомими алгоритмами складання розкладу у заданій постановці задачі є алгоритми Хелда-Карпа та Сахні-Горвіца. У наслідок того, що складність алгоритму Хелда-Карпа дорівнює $O\left(N \cdot 2^N\right)$ та є значно більшою ніж складність алгоритму Сахні-Горвіца – $O\left(N^2\right)$, то в основу методики формування плану доцільно покласти алгоритм Сахні-Горвіца.

Усе вище наведене дозволяє зробити висновок щодо необхідності розробки методики плану аналізу повідомлень оператором інформаційної мережі.

Сілко Д.О. (ВІТІ НТУУ „КПІ”)

МОДЕЛЬ МЕРЕЖЕВОГО ЕЛЕМЕНТА ЯК ОБ'ЄКТА ДІАГНОСТУВАННЯ

Питання забезпечення ЗСУ сучасними телекомунікаційними засобами протягом багатьох років залишалося пріоритетним напрямом технічної модернізації армії. Під час проведення командно-штабних навчань активно застосовувалися сучасні цифрові системи зв'язку та автоматизованого управління військами.

Невиконання функцій, для яких призначені цифрові телекомунікаційні засоби, як правило, зумовлено наявністю дефектів, які є фізичним явищем (обрив провідника, псування кристала та ін.). Дефекти, в свою чергу, формалізовано описують несправностями. Для успішної ідентифікації несправностей у телекомунікаційного засобу необхідно спочатку розглянути і визначити їх властивості як об'єктів діагностування. Насамперед це стосується математичних моделей і типів та моделей несправностей, що виникають у них.

З огляду на те, що цифровий телекомунікаційний засіб – складна система, складовими якої є елементи, компоненти і фрагменти мікропроцесорних пристроїв, змонтовані на друкарських платах (картриджах), загальну модель пропонується представити як багаторівневу (стратову), де кожному компоненту, фрагменту чи пристрою відповідає певний рівень (страт) деталізації. Узагальнена модель представлена множиною моделей $M = \{m_i\}$, $(i = 1, n)$, де модель m_i відображає поведінку станції на i-му рівні деталізації.

Якщо прийняти за k-й рівень модель, що описує мікропроцесорний пристрій цифрового телекомунікаційного засобу, яка відповідає платі об'єднувальній, то цей рівень можна представити алгоритмічною моделлю:

$$m_k = \langle X, Y, Q, F_\Delta, F_\lambda, P, S_A \rangle, \quad m_k \in m_{k+1},$$

де X, Y, Q – множина станів входів, виходів і внутрішніх станів вузлів плати об'єднувальної; F_Δ, F_λ – множини функцій переходів і виходів вузлів об'єднувальної плати; P – множина предикатів, за допомогою яких в алгоритмі реалізуються умовні переходи; S_A – граф-схема алгоритму, що має два типи вершин: операторні (обчислюють значення функцій на основі F_Δ і F_λ) і предикатні (перевіряють правильність заданого предиката і визначають наступну ділянку розрахункового шляху в алгоритмі).

При описі мікропроцесорного пристрої алгоритмічною моделлю домінуючу роль відіграють програмні засоби діагностування. Таким чином, процес діагностування спрощують, але місце несправності з точністю до компонента можна визначити при переході до функційної моделі на (k–1)-му рівні узагальненої моделі.

(k–1)-рівень представляють функційною моделлю:

$$m_{k-1} = \langle X, Y, Q, F_\Delta, F_\lambda, S_\Phi \rangle, \quad m_{k-1} \in m_k,$$

де де X, Y, Q – множини вхідних, вихідних і внутрішніх станів функційних вузлів, які відповідають компонентам картриджа; F_Δ, F_λ – множини функцій переходів і виходів функцій них вузлів картриджа; S_Φ – розмічена функцій на схема, відображена графом, з вершинами якого зіставлені функцій ні вузли, реалізовані як окремі компоненти, а з ребрами – зв'язки між ними. Моделювання на (k–1)-рівні дає змогу визначити місце прояву несправності з глибиною до компонента інтегральної схеми.

На основі наведених міркувань пропонується розробити вбудовані програмні засоби і методи контролю та діагностування для цифрових телекомунікаційних засобів. Такий підхід забезпечить здійснення контролю в процесі використання пристрою, простоту спрягання вмонтованих пристроїв діагностування з основною апаратурою.

Терещенко М.М. (ВІТІ НТУУ «КПІ»)

ОРГАНІЗАЦІЯ ПРИХОВАНОГО КАНАЛУ УПРАВЛІННЯ ОБ'ЄКТОМ ІНФОРМАЦІЙНОГО ДОСЛІДЖЕННЯ В ТЕХНОЛОГІЯХ МЕРЕЖЕВОЇ РОЗВІДКИ

Дана робота присвячена проблемі створення прихованого каналу управління об'єктом інформаційного дослідження в технологіях мережевої розвідки. Одним з перспективних напрямків розробки програмних продуктів є організація розробки методу для програмного додатку щодо непомітної роботи у комп'ютерних системах, а саме непомітній передачі даних та команд до об'єкту атаки, для нанесення максимальних деструктивних дій.

Для організації даного каналу серед наявних технологій мережевої розвідки було обрано:

- технологія перехоплення мережевого трафіку;
- методи сканування портів.

Орієнтована побудова додатку проводиться на технології клієнт-сервер. Основна ідея архітектури якої полягає в поділі мережевої програми на кілька компонентів, кожен з яких реалізує специфічний набір сервісів. Компоненти такого додатка можуть виконуватися на різних комп'ютерах, виконуючи функції як об'єкта так і джерелаатаки. Це дозволяє у будь який момент здійснити передачу даних тапідвищити надійність, безпеку і продуктивність мережевих додатків і мережі в цілому. Дана архітектура досить активно використовується у технологіях мережевої розвідки.

Як відомо прихований канал – це комунікаційний канал, який пересилає інформацію методом, який спочатку був для цього не призначений. Сторонні спостерігачі зазвичай не можуть виявити, що крім основного канал передачі даниxє ще додатковий. Тільки відправник і одержувач знають це. Серед наявних технологій побудови прихованого каналу управління було обрано канал на основі *IP* протоколу, а саме *ICMP*-тунель.

Прихований канал поверх *ICMP* встановлює приховане з'єднання між двома комп'ютерами за допомогою ехо-запитів та ехо-відповідей протокола ICMP. Дана техніка дозволює, наприклад, повністю тунелювати *TCP* трафік через ехо-запити і відповіді (*ping*). Технічно тунелювання через прихований канал *ICMP* відбувається способом введення будь-яких необхідних даних в ехо-пакет і його пересилки на віддалений компютер. віддалений компютер відповідає подібним чином заносячи відповідь у інший *ICMP* пакет і відсилаючи його назад.

Аналізуючи дані прийнятого пакету, можна сказати, що перші 14 байт представляють собою *Ethernet*-заголовок. Наступні 20 байт отриманого повідомлення – *IP*-заголовок. Далі йдуть 8 байт *ICMP*-заголовка, а далі йде саме *ICMP* повідомлення, у яке будуть внесені приховані для передачі дані.

Зі сказаного вище ясно, що ми можемо легко послати довільні дані на довільний хост шляхом впровадження даних в ехо-пакет. Тепер залишилося запустити на віддаленому хості демон, який вміє відповідати подібним чином, впроваджуючи відповідь в іншій *ICMP*-пакет і посилаючи пакет тому.

Щоб здійснити це, нам потрібно програмно реалізувати наступні дії:

- демон переглядає *ICMP*-пакети;
- після отримання пакету проводиться витяг його начинки (коректного запиту з боку атакуючого);
- на підставі начинки робляться необхідні дії;
- відсилається зворотний *ICMP*-пакет з відповіддю.

В якості математичного апарату використовується модель кінцевих автоматів.

Корнієнко А.Г. (КНУ)
Зур А.А. (КНУ)

РОЗВ'ЯЗАННЯ ЗАВДАНЬ ДИСКРЕТНОЇ ОПТИМІЗАЦІЇ МЕТОДОМ ГІЛОК ТА МЕЖ

У відповідності з [1 – 2], під завданням дискретної оптимізації розумітимемо завдання математичного програмування з кінцевою множиною допустимих рішень, що визначається такими виразами як

$$f(x^0) = \min_{x \in G} f(x),\ 0 \le |G| = N < \infty \tag{1}$$

або

$$f(x^0) = \max_{x \in G} f(x),\ 0 \le |G| = N < \infty, \tag{2}$$

$$N = 2^n, \tag{3}$$

де x^0 – оптимальне рішення; x – допустиме рішення; $f(x)$ – цільова функція; G – множина допустимих рішень; $|G|$ – число елементів множини G; n – число допустимих рішень.

У багатьох завданнях дискретної оптимізації умови дискретності виділені стосовно інших умов. Зважаючи на це розрізняють такі їх різновиди [1]:

1. Завдання цілочисленого лінійного програмування з оптимальними лінійним і цілочисленим рішеннями, що відрізняються одне від одного допустимими рішеннями та значеннями функцій, що оптимізуються.

2. Завдання цілочисленого лінійного програмування, для яких відсутні допустимі рішення для заданої множини допустимих рішень відповідного лінійного завдання.

Особливістю (1) – (2) є [1, 3], по-перше, не випуклість і не зв'язність області допустимих рішень. Це ускладнює їх розв'язання як регулярних завдань математичного програмування стандартними методами. Оскільки при цьому не виконуються характерні для них умови, а саме складно:

1. Визначити непустий окіл $G_x \subset G$ для кожного допустимого рішення $x \in G$.
2. Знайти локальний оптимум цільової функції на G через проблематичність ефективного формування необхідних і достатніх умов локальної оптимальності.
3. Забезпечити співпадання локального оптимуму з глобальним.

По-друге, для визначеного околу близькість двох його допустимих рішень $x^1 \in G$ та $x^2 \in G$ не вказує на близькість значень цільових функцій $f(x^1)$ та $f(x^2)$. Зважаючи на це, при розв'язанні завдань дискретної оптимізації вводяться штучні умови. Як наслідок, близькість двох допустимих рішень може бути оцінена тільки за значенням цільових функцій.

По-третє, розв'язання системи лінійних рівнянь та нерівностей в цілих додатних числах є складним обчислювальним завданням з класу *NP*. Його вирішення дозволяє підтвердити або спростувати існування допустимого рішення, пошук якого може бути проблематичним за аналогією з пошуком оптимального рішення.

По-четверте, якість алгоритму перебору варіантів оцінюється числом N допустимих рішень x, для яких визначається значення цільової функції $f(x)$. З огляду на (3), обсяги обчислень зростають при збільшенні кількості змінних. Тому перебір великої кількості варіантів ускладнений та визначається швидкодією електронно-обчислювальних машин.

Зважаючи на означені особливості завдань дискретної оптимізації для їх розв'язання здебільшого використовуються комбінаторні методи [3 – 4]. Зокрема, метод послідовного аналізу, метод гілок та меж, методи на основі послідовних схем, метод динамічного програмування, апроксимаційно-комбінаторний метод. Одним з найбільш

розповсюджених методів цієї групи є метод гілок та меж [5], який вперше запропоновано у 1960 р. у працях Ленд і Дойга для вирішення завдання цілочисленого лінійного програмування та згодом у працях Літтла, Мурті, Суіні, Керел для вирішення завдання комівояжера [6]. Це пов'язано з тим, що його використання не потребує великих обсягів машинної пам'яті, обчислювальна схема характеризується простотою, доступністю та скінченністю [7]. За основу методу гілок та меж взято ідею послідовного використання скінченності множини допустимих рішень завдання дискретної оптимізації для зменшення варіантів повного перебору направленим [1, 7]. Для цього множина допустимих рішень розбивається на підмножини та визначаються оцінки з метою обґрунтування вибору підмножин без допустимих й оптимальних рішень. Тому такий підхід дозволяє скоротити перебір варіантів розв'язання завдань (1) – (2) дискретної оптимізації за вісім етапів [1, 6]:

1. Визначення оцінок

Нехай $G' \subset G$, тоді $\varphi(G')$:

– нижня оцінка (1), якщо для будь-якого $x \in G$ виконується нерівність

$$f(x) \geq \varphi_{\min}(G'); \tag{4}$$

– верхня оцінка (2), якщо для будь-якого $x \in G$ виконується нерівність

$$f(x) \leq \varphi_{\max}(G'). \tag{5}$$

2. Розбиття множини G допустимих рішень на підмножини

Позначимо множину допустимих рішень $G = G^0$ та розіб'ємо G^0 на r_1 підмножин

$$G_1^1, G_2^1, .., G_{r1}^1;\ G^0 = \bigcup_{i=1}^{r1} G_i^1;\ G_i^1 \cap G_j^1 = \varnothing,\ i \neq j \in (1, r_1).$$

На k кроці для розбиття на підмножини вибираємо одну з множин $G_1^k, G_2^k, .., G_{r_k}^k$

$$G_s^k = \bigcup_{t=1}^{r_k} G_{s,t}^k.$$

Завдяки цьому виразимо множину G_s^k через підмножини $G_{s,t}^k$ та виділимо серед них ті

$$G_1^{k+1}, G_2^{k+1}, .., G_{r_k}^{k+1}, \tag{6}$$

що потребують розбиття. Як наслідок, отримаємо список (6) завдань на розгалуження (див., наприклад [6], рис. 1).

3. Перерахування оцінок

Якщо $G_1 \subset G_2$, то

$$(1): \min_{x \in G_1} f(x) \geq \min_{x \in G_2} f(x)$$

або

$$(2): \max_{x \in G_1} f(x) \leq \max_{x \in G_2} f(x).$$

Тому, в процесі розгалуження підмножини $G' \subset G$ на підмножини

$$G_1', G_2', .., G_s';\ G' = \bigcup_{i=1}^{s} G_i';\ G_i' \cap G_j' = \varnothing,\ i \neq j \in (1, s)$$

будемо вважати, що

$$(1): \varphi(G_i') \geq \varphi_{\min}(G') \tag{7}$$

або

$$(2): \varphi(G_i') \leq \varphi_{\max}(G'). \tag{8}$$

До того ж можливе покращення оцінки (7) або (8) завдяки виконанню строгої нерівності для деяких номерів i_0, тобто

$$(1): \varphi(G_{i_0}') > \varphi_{\min}(G')$$

або

$$(2): \varphi(G'_{i_0}) < \varphi_{\max}(G').$$

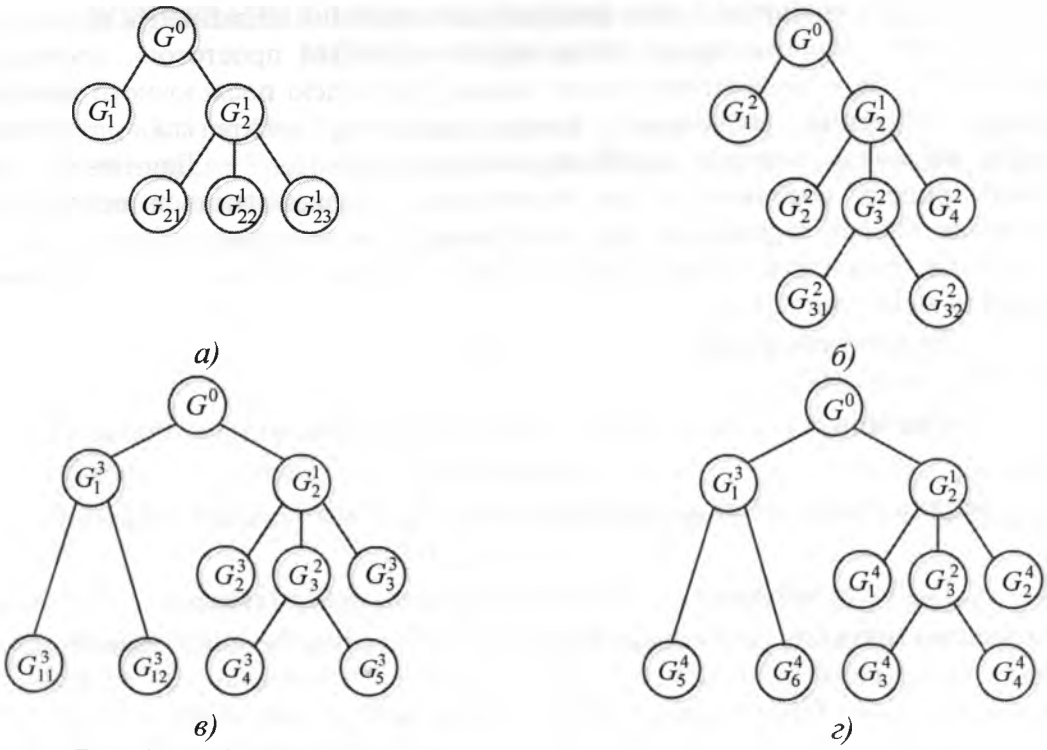

Рис. 1. Варіанти розбиття множини допустимих рішень на підмножини

4. Визначення допустимих рішень

Якщо на k та $(k+1)$ кроках знайдено допустимі рішення x^k та x^{k+1}, то при виконанні нерівності

$$(1): f(x^{k+1}) < f(x^k)$$

або

$$(2): f(x^{k+1}) > f(x^k)$$

допустиме рішення x^{k+1} буде найкращим.

5. Перевірка оптимальності допустимого рішення

Нехай

$$G = \bigcup_{i=1}^{s} G_i$$

і допустиме рішення x^k належить деякій підмножині G_v. Тоді, якщо справедлива умова

$$(1): f(x^k) = \varphi(G_v) \le \varphi(G_i)$$

або

$$(2): f(x^k) = \varphi(G_v) \ge \varphi(G_i),$$

то x^k є оптимальним рішенням завдань (1), (2).

6. Оцінювання точності наближених рішень

Нехай

$$(1): \varphi_0 = \min_{1 \le i \le s} \varphi(G_i), G_i \in G$$

або

$$(2): \varphi_0 = \max_{1 \le i \le s} \varphi(G_i), G_i \in G.$$

Тоді справедлива така нерівність

$$(1): \varphi_0 \le f(x^0) \le f(x^k)$$

або

$$(2): \varphi_0 \ge f(x^0) \ge f(x^k).$$

При цьому модуль різниці

$$\Delta = \left| f(x^k) - \varphi_0 \right|$$

буде оцінкою відхилення найкращого допустимого рішення x^k від оптимального x^0 для (1) та (2). Тому для розгалуження доцільно вибирати множину з мінімальним значенням (4) або максимальним значенням (5). Завдяки цьому можливе зменшення величини Δ та отримання точного та ε-оптимального рішення ($\varepsilon > 0$) при $\Delta \le \varepsilon$.

7. Правило відсівання

Якщо x^k – найкраще допустиме рішення, то при виконанні умови

$$(1): \varphi(G_r) > f(x^k) \tag{9}$$

або

$$(2): \varphi(G_r) < f(x^k) \tag{10}$$

множина G_r відсівається. Це обумовлено відсутністю в ній оптимальних рішень тому, що з урахуванням (4), (5) отримаємо

$$(1): f(x) \ge \varphi(G_r) \ge f(x^k) \ge f(x^0)$$

або

$$(2): f(x) \le \varphi(G_r) \le f(x^k) \le f(x^0).$$

При цьому використання умов (9), (10) дозволяє запобігти відсіванню підмножин, що містять оптимальне рішення x^0. Водночас для отримання хоча б одного оптимального рішення x^0 на практиці доцільно брати за основу такі умови

$$(1): \varphi(G_r) \ge f(x^k)$$

або

$$(2): \varphi(G_r) \le f(x^k).$$

8. Скінченність алгоритму обумовлена скінченністю множини допустимих рішень G.

Зважаючи на означені етапи скорочення перебору варіантів розв'язання завдань дискретної оптимізації (1) або (2), ефективність методу гілок та меж визначається числом розв'язаних завдань [1, 6 – 7] за два етапи [1]:

1. Отримання оптимального рішення.
2. Доведення оптимальності отриманого рішення.

Як наслідок, кількість вирішених завдань при отриманні рішення може бути значно меншою за кількість вирішених завдань при доведенні його оптимальності. Для цього множина G^0 спочатку записується в список завдань для розгалуження. Після її виконання список змінюється шляхом запису до нього множин $G_i, i \in (1, s)$. Така зміна відбувається на кожному кроці процесу розгалуження. В результаті цього отримуємо список завдань-кандидатів.

Загальна схема розв'язання завдань методом гілок та меж включає такі кроки (див., наприклад [1], рис. 2):

1. Запис вихідного завдання у список завдань-кандидатів.
2. Якщо у списку відсутні завдання-кандидати, то завдання вирішене. Інакше необхідно перейти до кроку 3.
3. Вибір множини для розгалуження.
4. Розгалуження та модифікування списку завдань-кандидатів.
5. Визначення оцінок та знаходження найкращих допустимих рішень.

6. Відкидання множин без оптимальних рішень та перевірка ознаки оптимальності.
7. Перехід до кроку 2.

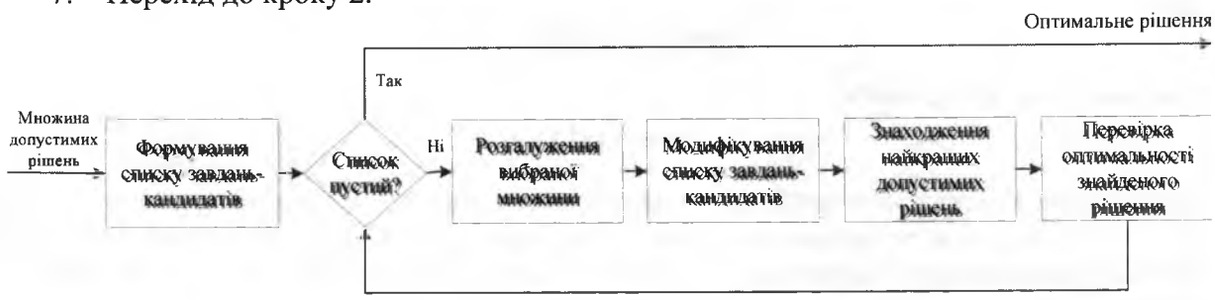

Рис. 2. Загальна схема розв'язання завдань методом гілок та меж

Таким чином, при розв'язанні завдань дискретної оптимізації необхідно враховувати характерні для них особливості. Зокрема, не випуклість, не зв'язність області допустимих рішень; оцінювання близькості допустимих рішень тільки за значенням цільових функцій; обчислювальну складність визначення оптимальних рішень. Як наслідок, складно знайти окіл для кожного допустимого рішення, забезпечити рівність локального оптимуму з глобальним та проблематично сформувати необхідні і достатні умови локальної оптимальності. Зважаючи на це, метод гілок та меж дозволяє знайти оптимальне рішення шляхом зменшення кількості варіантів повного перебору. Його використання не потребує великих обсягів машинної пам'яті, обчислювальна схема характеризується простотою та доступністю. Це стало можливим завдяки скінченності множини допустимих рішень завдання (1) або (2) дискретної оптимізації методом гілок та меж, ефективність застосування якого визначається числом розв'язаних завдань.

ЛІТЕРАТУРА:

1. Сигал И.Х. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы: учеб. пособие: [изд. 2-е, испр.] / И.Х. Сигал, А.П. Иванова. – М.: ФИЗМАТЛИТ, 2003. – 240 с.
2. Сухарев А. Г. Курс методов оптимизации: учеб. пособие / А.Г. Сухарев, А.В. Тихомиров, В.В. Федоров. – 2-е изд. – М.: ФИЗМАТЛИТ, 2005. – 368 с.
3. Ковалев М.М. Дискретная оптимизация (целочисленное программирование): [изд. 2-е, стереотипное] / М.М. Ковалев. – М.: Едиториал УРСС, 2003. – 192 с.
4. Зайченко Ю.П. Исследование операций: учеб. пособие для студентов вузов: [изд. 2-е, перераб. и доп.] / Ю.П. Зайченко. – К.: Высшая школа. Головное издательствово, 1979. – 392 с.
5. Волков И.К. Исследование операций: учеб. для вузов / И.К. Волков, Е.А. Загоруйко // Под ред. В.С. Зарубина, А.П. Крищенко. – М.: Издательство МГТУ им. Н.Э. Баумана, 2000. – 436 с.
6. Корбут А.А. Дискретное программирование / А.А. Корбут, Ю.Ю. Финкельштейн. – М.: Издательство «Наука», Главная редакция физико-математической литературы, 1969. – 368 с.
7. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации / И.В. Сергиенко. – К.: Наукова думка, 1988. – 472 с.