

нечітких множин [3, 4]. Вибір значень елементів множини завжди пов'язаний з ризиком того, що обрані значення показників не забезпечують необхідного (заданого, прогнозованого) рівня безпеки. Наслідком цього є можливість здійснити ефективну атаку на інформаційні ресурси підприємства. Пошук наочного представлення знань приводить до систем на основі нечіткої логіки, яка й забезпечує представлення словесно інтерпретованих знань.

Для забезпечення можливості прийняття рішення при не тільки кількісних, але й якісних характеристиках комплексної системи захисту інформації підприємства, доцільне застосування нечіткого ієрархічного дерева, на основі якого здійснюється побудова бази знань СКЗІ. Такий комплексний підхід надає можливість отримати на етапі проектування оптимальний варіант системи інформаційної безпеки підприємства, а також доцільний для застосування у системі [7].

Список літератури

1. Доренський О.П. Особливості класифікації загроз безпеці інформації сучасної інформаційно-телекомунікаційної системи // Збірник наукових праць Кіровоградського національного технічного університету. – Вип. 20. – Кіровоград: КНТУ, 2008. – С. 155-161.
2. Галатенко В.А. Основы информационной безопасности. – М.: "ИУИТ", 2011. – 208 с.
3. Дудат'єв А.В. Проектування системи безпеки інформаційних ресурсів підприємства. / Методи та засоби кодування, захисту й ущільнення інформації. – Вінниця: ВНТУ, 2007. – С. 75-76.
4. Домарев В.В. Безопасность информационных технологий. Системный подход. – К.: ООО "Тид "ДС", 2004. – 992 с.
5. Смірнов О.А. Основы захисту інформації: Навч. пос.. / А.О. Смірнов, Л.Г. Віхрова, С.І. Осадчий, В.Ю. Ковтун, Є.В. Мелешко. – Кіровоград: РВЛ КНТУ, 2011. – 322 с.
6. ISO/IEC 15408-1:2009 "Information technology. Security techniques. Evaluation criteria for IT security" [Електронний ресурс]. – Режим доступу: www.iso.org.
7. Загорський Р.М. Програмне забезпечення побудови моделі захисту інформації телекомунікаційної системи. / Загорський Р.М., Доренський О.П. // Матеріали XLIV наукової конференції студентів та магістрантів КНТУ – Кіровоград: КНТУ, 2011. – с. 78.

Платформа CLR в середовищі ОС Windows

О.О. Іванченко, студент,

В.А. Бісюк, викладач

Кіровоградський національний технічний університет

Common Language Runtime (скорочено CLR — «загальне середовище виконання мов») — це компонент пакету Microsoft .NET Framework, віртуальна машина, на якій виконуються всі мови платформи .NET Framework.

CLR транслює початковий код в байт-код на мові IL, реалізація компіляції якого компанією Microsoft називається MSIL, а також надає MSIL-програмам (а отже, і програмам, написаним на мовах високого рівня, що підтримують .NET Framework) доступ до бібліотеки класів .NET Framework, або так званої .NET FCL.

Середовище CLR є реалізацією специфікації CLI (англ. Common Language Infrastructure), специфікації загальномовної інфраструктури, компанії Microsoft.

Віртуальна машина CLR дозволяє програмістам забути про багато деталей щодо конкретного процесору, на якому виконуватиметься програма. CLR також забезпечує такі важливі служби як: керування пам'яттю; керування потоками; обробка виключень; збірка сміття; безпека виконання.

Компонент Common Language Runtime розташовується над сервісами операційної системи, якою в даний час є операційна система Windows, але надалі такою може бути практично будь-яка програмна платформа.

Основне призначення CLR - виконання програм, дотримання всіх програмних залежностей, керування пам'яттю, забезпечення безпеки, інтеграція з мовами програмування тощо. Середовище виконання забезпечує безліч сервісів, що полегшують створення та впровадження програм, і істотно покращує надійність останніх.

Розробники не взаємодіють з Common Language Runtime безпосередньо: всі сервіси надаються уніфікованою бібліотекою класів, яка розташовується над CLR. Ця бібліотека містить більше 1000 класів для вирішення різних програмних завдань - від взаємодії з сервісами операційної системи до роботи з даними і XML-документами.

Common Language Runtime забезпечує середовище виконання .NET-додатків. Серед наданих цим середовищем функцій слід відзначити обробку виключних ситуацій, забезпечення безпеки, засоби налагодження підтримки версій. Всі ці функції доступні з будь-якої мови програмування відповідно до специфікації Common Language Specification. Microsoft надає три мови програмування, здатних використовувати CLR: Visual Basic, NET, Visual C#. NET і Visual C++ With Managed Extensions. Крім того, ряд третіх фірм працює над .NET-версіями таких мов програмування, як Perl, Python і COBOL.

Компільований компілятором код для CLR називається керованим кодом (managed code). Керований код користується перевагами середовища виконання і крім власне коду містить метадані, які створюються в процесі компіляції і містять інформацію про типи, члени і посилання, що використовуються в коді. Метадані використовуються середовищем виконання.

Середовище виконання також стежить за часом життя об'єктів. В COM/COM+ з цією метою використовувалися спеціальні лічильники (reference counter). У CLR теж використовуються лічильники, а видалення об'єктів з пам'яті відбувається за допомогою процесу, званого збіркою сміття (garbage collection).

Common Language Runtime також задає загальну систему типів, яка використовується всіма мовами програмування. Це означає, наприклад, що всі мови програмування будуть оперувати цілочисельними даними або даними з плаваючою точкою єдиного формату і єдиної довжини, а уявлення рядків теж будуть єдиними для всіх мов програмування. За рахунок єдиної системи типів досягається більш проста інтеграція компонентів і коду, написаних на різних мовах програмування. На відміну від COM-технології, також заснованої на наборі стандартних типів, але які подаються в бінарному вигляді, CLR дозволяє виконувати інтеграцію коду (який може бути написаний на різних мовах програмування) в режимі дизайну, а не в режимі виконання.

Після компіляції керований код містить метадані, що описують сам компонент, а також компоненти, що використовувалися для створення коду. Середовище виконання перевіряє, чи доступні всі необхідні ресурси. Використання метаданих дозволяє відмовитися від необхідності зберігати інформацію про компоненти в реєстрі. Отже, при перенесенні компонента на інший комп'ютер більше не потрібно реєструвати цей компонент, а видалення компонента зводиться до простого видалення збірки, яка його містить.

Список літератури

1. Інтернет-енциклопедія Wikipedia (www.wikipedia.org).
2. NET Common Language Runtime Unleashed. 2-volume set, Kevin Burton, Sams, 2002