УДК 361.3.06

МОДЕЛЬ БЕЗПЕКИ СИСТЕМ УПРАВЛІННЯ КОРПОРАТИВНОЮ ІНФОРМАЦІЄЮ НА БАЗІ КОНТРОЛЮ ВИКОНАННЯ КОДУ ПЛАТФОРМИ .NET SECURITY FRAMEWORK

В.М. Богуш,С.О. Спасітєлєва
Державний університет інформаційно-комунікаційних технологій,
Вул. Солом'янська, 7, 03110, Київ, Україна

Розглядається модель безпеки систем управління корпоративним вмістом при програмуванні на платформі .NET Security Framework. Аналізуються можливості компонентно-орієнтованої моделі контролю доступу до виконуваного коду програми, виходячи із його ідентифікації.

Концепція управління корпоративним вмістом - Enterprise Content Management (ECM), яка визначається як інтегрований підхід до управління корпоративними документами й Web-змістом, має багато переваг і тому набула широкої популярності. ECM-система інтегрує контентно- і процесно-орієнтовані технології підприємства і забезпечує загальну інфраструктуру для управління інформацією. Сутність такого інфраструктурного підходу полягає в тому, що корпоративний вміст може належати різним додаткам від різних виробників, може бути доступним для різних додатків і вільно поширюватися між ними. Це забезпечує здатність інтеграції ECM-систем з зовнішніми ERP-системами, офісними додатками, сховищами, системами електронного документообігу тощо. Інтеграція може бути виконана за допомогою використання об'єктно-орієнтованих інтерфейсів. В цей час відсутні чіткі погляди на способи створення відкритої ECM-інфраструктури, що інтегрують спеціалізовані системи від різних виробників у межах підприємства [1]. Додавання до ECM-систем спеціалізованих модулів значно розширює їхні функціональні можливості, але одночасно підвищує небезпеку їх використання. Для забезпечення високого ступеню інтеграції ECM-системи з різними програмними додатками і реалізації функціональних можливостей, пов'язаних з безпекою, пропонується використовувати можливості та переваги платформи .NET Security Framework.

Концепція безпеки платформи .NET Security Framework

Платформа .NET Security Framework, яка забезпечує незалежне від мови середовище розробки розподілених прикладних програм, робить ці програми могутнішими з точки зору безпеки, а сам процес розробки простішим. Платформа підтримує дві концепції безпеки створення надійних програм:

1. Безпека на базі ідентифікації користувача та «ролі», до якої він належить. При цьому система захисту фокусує увагу на правах користувачів і дозволяє або забороняє дії, виходячи із ідентифікації поточного користувача програми. Відповідна традиційна модель безпеки будується на концепції реєстрації користувача в системі та на контролі над виконуваними процесами.

2. Безпека на базі контролю доступу до коду Code Access Security (CAS). На відміну від вищенаведеного традиційного підходу, CAS дозволяє контролювати виконання дій, виходячи із ідентифікації власне коду, а не ідентифікації користувача, який цей код виконує. CAS дозволяє вирішувати, які дії дозволено виконувати коду програми.

Платформа підтримує обидві концепції і надає можливості накладати обмеження на дії користувача та на виконання коду [2]. Реалізація CAS дозволяє забороняти деяким секціям коду отримувати доступ до певних файлів та інших ресурсів або блокувати виконання стороннього коду. Це є особливо корисним для загальнодоступних Web-вузлів та Web-сервісів, на яких нереально мати облікові записи та блокування ресурсів для великої кількості користувачів, або коли потрібно виконувати код, створений невідомим розробником. Предметом нашого розгляду буде модель CAS, яка використовує особливості побудови середовища розробки.

Середовище розробки .NET Security Framework складається з двох взаємопов'язаних частин: єдиного середовище виконання незалежного від мови («віртуальна машина») - Common Language Runtime (CLR) та бібліотеки класів (див. рис. 1). Використання можливостей CLR бібліотечних функцій дозволяє програмісту ефективно вирішувати проблеми безпеки [3].Компіляція коду для .NET являє собою компіляцію для CLR, незалежну від мови програмування. Керований код спочатку компілюється у мову проміжного рівня MSIL (Microsoft Intermediate Language). Код MSIL далі компілюється в виконуваний код в період виконання, «на льоту» віртуальною машиною. При цьому CLR отримує можливість перевіряти код перед його виконанням, прочитавши його метадані, і забезпечує безпеку на рівні виконання додатку. При цьому гарантується, що додаток не може нашкодити користувачу або порушити функціонування ОС. Це досягається за рахунок того, що при компоновці прикладної програми на відміну від звичайних файлів створюються файли, які містять не тільки керований код але і метадані. Такі виконувані файли (portable executable -PE) називаються зборками (assemblies). Зборку можна розглядати як готовий до виконання додаток. Метадані описують вміст та атрибути коду (типи даних, класи, методи та властивості класів). Це означає, що кожна зборка у форматі EXE- або DLL-файла є самостійним об'єктом і може без додаткових даних бути розміщеною у системному реєстрі. При цьому CLR може аналізувати об'єкт стосовно його безпеки, прочитавши його метадані. CAS базується на тому, що можна визначити рівень довіри до зборок та обмежити для цього коду виконання певних операцій.

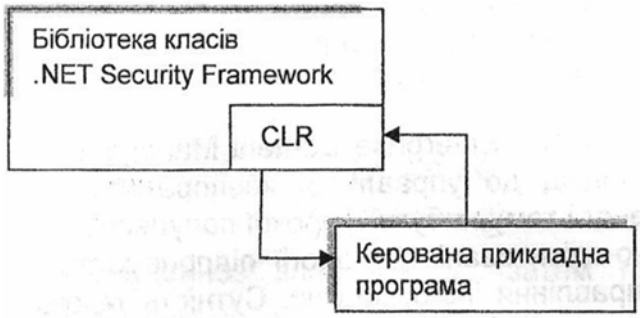

Рис. 1. Компоненти платформи .Net Security Framework

Безпека на базі контролю виконання коду додатків - Code Access Security (CAS)

Безпека виконання коду забезпечується такими можливостями :

•Верифікація коду в процесі виконання, тобто перевірка діапазонів адрес та контроль типів, що робить неможливим руйнування даних у пам'яті та не дозволяє небезпечних перетворень типів даних і прямих маніпуляцій з даними в пам'яті через вказівник. Таким чином, керований код з контролем типів запобігає атакам, пов'язаним з переповненням буферу, «перекриттям стеку» та не дозволяє вставку довільного (небезпечного, вірусного) коду.

Верифікація коду може здійснюватися віртуальною машиною CLR автоматично.

- Контроль виконання коду, який будується на механізмі дозволів і регулюється політикою безпеки середовища виконання CLR на базі «свідоцтв». Це запобігає розповсюдженню потенційно небезпечних програм. Політика безпеки може застосовуватися тільки до верифікованого коду з контролем ТИПІВ.

Таким чином маємо дворівневу модель безпеки додатків на базі контролю виконання коду. Якщо на етапі верифікації коду все ж таки відбувається вставка у зборку несанкціонованого коду, тоді на другому етапі виконується захист з використанням політики безпеки для відповідної зборки.

Зборка не може виконати дії не дозволені політикою безпеки, що задається для кожної зборки. Завантажена зборка містить додаткову інформацію, яка називається свідоцтвом безпеки (security evidence). CLR використовує цю інформацію для прийняття рішення про надання дозволу на виконання певних повноважень коду.

Свідоцтво, яке використовується для авторизації в середовищі .NET може містити такі компоненти:

- Strong Name - криптографічно стійкий спосіб ідентифікації зборки (цифровий підпис).
- Zone - зона, з якої зборка була отримана, наприклад, локальний комп'ютер, Інтернет або локальна мережа.
- URL - URL, з якого завантажена зборка.
- Site - місцезнаходження сайту, на якому знаходиться зборка.
- Hash - криптографічний хзш-код зборки.
- Application Directory - директорія, із якої завантажена зборка.
- Publisher certificate - сертифікат зборки (приватне ім'я й версію зборки, назву виробника зборки).

Виробник зборки може також додати й іншу інформацію, яка буде враховуватися в тому випадку, якщо політика безпеки комп'ютера налаштована на використання приватної інформації.

Дані, що описують інформацію про зборку, розрізняються по надійності. Наприклад, строге ім'я або ім’я виробника мають більший пріоритет, тому що їх важче підробити, у порівнянні із зоною зборки й адресою сайту. Надавати доступ на основі недостатньо надійних даних про зборку є небезпечним, але якщо поєднати всі дані про зборку, то можна досягти гарного результату.

Свідоцтво використовується для внесення зборки до певної групи коду і надання їй певних дозволів у залежності від політики безпеки. Дозвіл - це об’єкт, що описує права та привілеї зборок, які належать до певної групи коду, на доступ до зазначених ресурсів та виконання певних дій. Наприклад, до дій, що захищаються дозволами доступу, відносяться такі, як читання й запис файлу, доступ до даних операційної системи або доступ до даних, які містяться в базі даних.

Політика безпеки - це набір правил, який конфігурується адміністратором і використовується середовищем CLR для прийняття CAS-рішення. Фактично політика задає відповідність між свідоцтвами та дозволами для кожної зборки. Ще однією можливістю контролю рівня безпеки є те, що зборка може програмно запросити дозвіл на виконання певних дій. У цьому випадку середовище виконання CLR приймає рішення на базі поточної політики безпеки і видає дозвіл або відмову.

Політика безпеки має багаторівневу структуру і може бути задана на рівні підприємства, комп’ютера, прикладної програми або користувача. В таблиці 1 наведено опис рівнів політики безпеки. Політика підприємства має пріоритет над політикою кожного комп’ютера, а політика комп’ютера - над будь-якою політикою прикладної програми або користувача. Якщо між дозволами на певному рівні існує

конфлікт, тоді обирається більш обмежений варіант. Остаточний набір дозволів визначається у результаті суми всіх дозволів від кожної зборки.

Таблиця 1. Рівні безпеки

Рівень безпеки	Опис
Мережа підприємства (Enterprise level)	Рівень контролюється адміністратором мережі й містить групи коду, які можуть бути призначені будь-якому керованому коду у всій корпоративній мережі
Комп'ютер (Machine)	Рівень контролюється адміністратором комп'ютера й містить групи коду, які можуть бути призначені будь-якому керованому коду, виконуваному на даному комп'ютері
Користувач (User)	Рівень контролюється користувачем комп'ютера й містить групи коду, які можуть бути призначені будь-якому керованому коду, виконуваному від імені даного користувача
Домен додатка (Application Domain)	Додатковий рівень, що дозволяє ізолювати, вивантажувати й обмежувати виконуваний керований код

Середовище виконання CLR також має стандартну політику безпеки й використовує її набір дозволів. Настроювання безпеки кожного комп'ютера індивідуальні. Вони міняються залежно від користувача, адміністратора комп'ютера або адміністратора мережі. Взаємодія із системою безпеки під час виконання здійснюється двома способами: імперативним і декларативним. Декларативний спосіб взаємодії здійснюється за допомогою атрибутів, імперативний - за допомогою об'єктів класів у додатку. ^

Для управління політикою безпеки платформи .NET Security Framework можна використовувати засіб конфігурування .NET Framework Configuration або утиліту політики безпеки CAS.

Класи бібліотеки .NET Security Framework Бібліотека класів NET Framework використовує ієрархічну схему іменувань. Зв'язні між собою класи групуються в простір імен, що спрощує їх пошук та створення посилань. System - це ім'я кореневого простору імен для головних класів. На рисунку 2 надана характеристика та ієрархія просторів імен, у яких міститься опис класів для реалізації функцій безпеки набір класів, визначених в просторі імен System.Security. До базових класів системи безпеки належать класи: CodeAccessPermission, SecurityManager, SecurityElement, PermissionSet, NamedPermissionSet, VerificationExeption Наприклад, клас System.Security.SecurityManager використовується для задавання відповідності між свідоцтвами групи і дозволами, що визначає реальну політику безпеки. Клас System. Security. CodeAccessPermission також є визначеним в просторі імен System.Security і використовується як базовий для множини класів дозволів. В просторі імен System.Security.Policy є визначеними класи свідоцтв та умов членства: ApplicationDirectory, CodeGroup, Evidence, Hash, PolicyLevel, Site, Url, Zone

StrongName, Publisher. Наприклад, для ідентифікації зборок використовується клас System.Security.Policy.Evidence, який інкапсулює інформацію про свідоцтва для визначення характеристик зборки.

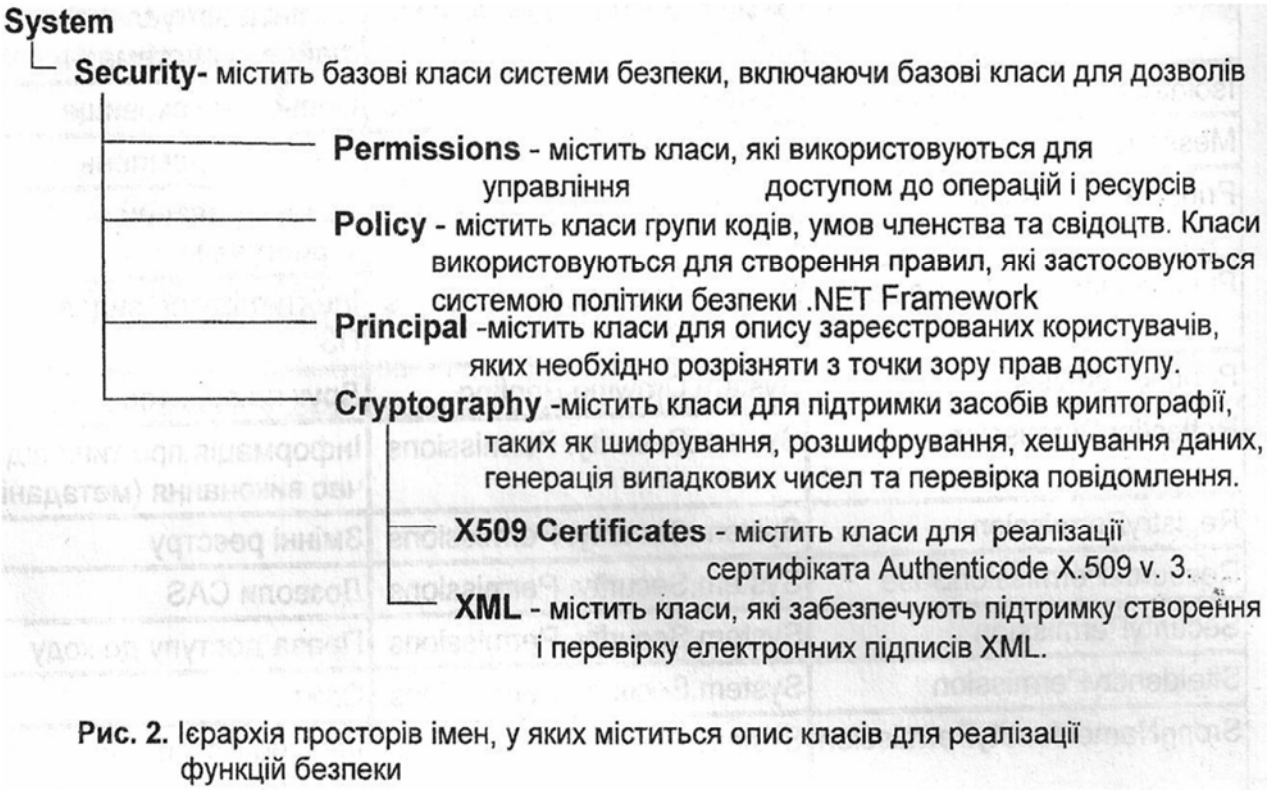

Рис. 2. Ієрархія просторів імен, у яких міститься опис класів для реалізації функцій безпеки

В просторі імен System.Security.Principal є визначеними класи зареєстрованих користувачів: GenericPrincipal, GenericIdentity, WindowsIdentity В просторі імен System.Security.Permissions визначається множина похідних класів, які успадковують від класу System.Security.CodeAccessPermission і використовуються для перевірки дозволів зборок. Середовище .NET Security Framework має багато убудованих похідних класів дозволів, що автоматично перевіряють код програми і відповідні дозволи. У таблиці 2 перераховано всі вбудовані класи дозволів, визначені в різних просторах імен.

Таблиця 2. Убудовані дозволи .NET Security Framework

Клас, що представляє дозвіл	**Простір імен**	**Ресурс що захищається**
DnsPermission	System. Net	Служби DNS (Domain Namespace Services)
DBDataPermission	System. Data.Common	Бази даних
EnvironmentPermission	System.Security. Permissions	Змінні операційної системи та середовища користувача
EventLogPermission		Журнал подій

FileDialogPermission	System.Security. Permissions	Діалоги відкриття, збереження файлів у інтерфейсі додатка
FileIOPermission	System.Security. Permissions	Доступ до файлів та папок
IsolatedStorgeFilePermission	System.Security. Permissions	Закрита віртуальна файлова система
IsolatedStoragePermission	System.Security. Permissions	Ізольоване сховище
MessageQueuePermission	System.Messaging	Черги повідомлень
PrincipalPermission	System.Security. Permissions	Зареєстрований користувач
PublisherIdentityPermission	System.Security. Permissions	Ідентифікатор видавця ПЗ
PrintingPermission	System.Drawing.Printing	Друк на принтер
ReflectionPermission	System.Security. Permissions	Інформація про типи під час виконання (метадані)
RegistryPermission	System.Security. Permissions	Змінні реєстру
ResourcePermissionBase	System.Security. Permissions	Дозволи CAS
SecurityPermission	System.Security. Permissions	Права доступу до коду
SiteIdentityPermission	System.Security. Permissions	Сайт
SrongNameIdentityPermission	System.Security. Permissions	Ідентифікатор для строгих імен, цифровий підпис
SocketPermission	System.Net	З'єднання з комп'ютером через порти
UIPermission	System.Security. Permissions	Користувальницький інтерфейс, буфер обміну
UrlIdentityPermission	System.Security. Permissions	URL
ZoneIdentityPermission	System.Security. Permissions	Зони
WebPermission	System.Net	З'єднання з комп'ютерами через web-інтерфейс

На додаток до убудованих дозволів можна створити власні дозволи, використовуючи клас CustomPermission.

Також платформа надає могутню підтримку всіх задач, пов'язаних з криптографією. Бібліотека криптографічних класів забезпечує доступ до більшості криптографічних алгоритмів: DSA, RSA, DES, RC2, KeyedHashAlgorithm, MD5, SHA, SHA256, SHA384, SHA512. Криптографічні класи реалізовані в просторі імен System. Security.Cryptography.

Висновки

Модель безпеки .NET Security Framework будується поверх моделі безпеки операційної системи, також вона може взаємодіяти з функціями безпеки різних серверних додатків, таких як SQL Server або IIS (Internet Information Server). Таким

чином, характеристики безпеки прикладних програм .NET Security Framework залежать від конфігурації власних функцій безпеки, від програмування компонентів прикладної програми, від налаштувань ОС, мережевого оточення. Безпека середовища виконання CLR може налаштовуватися детальніше ніж безпека ОС, що дозволяє делегувати контроль безпеки керованого коду в CLR. Інфраструктура .NET Security Framework забезпечує засоби для створення захищених прикладних програм шляхом створення середовища, у якому кожну програму можна контролювати на базі CAS, ідентифікації користувачів, криптографічних методів та політики безпеки середовища CLR. Таким чином .NET Security Framework забезпечує гнучку надбудову над традиційною системою безпеки і дозволяє створювати надійні, гарно захищені інтегровані системи управління на базі багаторівневої моделі захисту.

Література

1. *Дж. Д. Саттон* Корпоративний документооборот: принципы, технологи, методология внедрения / Пер. с англ. ~ Санкт-Петербург.: «БМикро», 2002. -448 с.
2. *Торстейнсон П., Ганеш Г.А.* Криптография и безопасность в технологии .NET/ Пер. с англ. - М.: БИНОМ. Лаборатория знаний, 2007. -479 с.
3. *Дж. Просиз* Программирование для Microsoft .NET/ Пер. с англ. - М.: Издательско-торговий дом «Русская редакция», 2003. - 704 стр.