О. Грохольский
Науковий керівник – асист. М.Л. Навитка

АНАЛІЗ СТРУКТУРИ ОБ'ЄКТНИХ БРОКЕРІВ ОРГАНІЗАЦІЇ ДОСТУПУ

Кількість об'єків сучасних комп'ютерних систем (КС) є великою та потребує захисту. Одним із способів захисту апаратного та програмного забезпечення є процедура організації безпечного доступу до цих об'єктів.

З метою пришвидшення організації доступу використовується ідеологія політики захисту (від кого і які дані повинні захищатись) і механізму її реалізації. В деяких системах захист реалізується за допомогою ідеології монітора повідомлень [1] і полягає в тому, що при спробі доступу до деякого ресурсу система на початку запитує монітор перевірити законність даного доступу. Монітор звертань переглядає таблиці політики і приймає рішення стосовно надання чи відмови в доступі.

Реалізація оточення в якому працює монітор звертань є різним. Серед найбільш вживаних розглядаються домени захисту, списки доступу, політики груп і ін. [1, 2]. Усіх їх об'єднує одна спільна риса – існування персоніфікованого запису, на основі якого приймається рішення про доступ. Зазвичай такий запис є унікальним для кожного користувача чи об'єкта, які запитують доступ і має назву policy.

Основною метою статті є аналіз об'єктних брокерів доступу для реалізації ідеології multi policy в системах з політикою груп.

Для досягнення цієї мети завдання необхідно вирішити такі завдання:

1) аналіз системи персоніфікованого безпечного доступу одночасно до багатьох об'єктів;
2) аналіз системи об'єктних брокерів, які призначення для вирішення завдання надання чи відмови в доступі.

На рис.1 наведена схема користувацької системи безпеки доступу до багатьох об'єктів (user multi policy, *UMP*), якою пропонується замітити існуючу (simple policy, *SP*). Уособленням *SP* в схемі *UMP* є *USP* (user simple policy). Відповідність *UMP* і *USP* верхнім індексом, наприклад для наведеного на рис.1 UMP^1 відповідає таблиця USP^1.

Фактично UMP^i є таблицею вказівників USP^i *ptr*[*j*] на об'єкти безпеки (Policy objects, *PO*), для "користувачів" із системним індексом *i*. Тут користувачами, окрім звичайних зареєстрованих в системі користувачів, можуть виступати системні об'єкти, сервіси, тощо.

Довжина таблиці USP^i є довільною, що дає можливість в одній системі володіти багатьма об'єктами *SP* з різними правами доступу.

PO ^{i} *j* є фізично існуючими об'єктами безпеки. Саме вони реалізовують ідеологію доступу до інших об'єктів (надалі сервісів) в системі. При цьому *USP* ^{i} *ptr*[*j*] забезпечують лише доступ до цих об'єктів. Відповідність *USP* ^{i} *ptr*[*j*] і *PO* ^{i} *j* визначається індексом *j*.

Відзначимо, що кожен *PO* ^{i} *j* належить виключно лише одному *UMP* ^{i}. Більше того самостійно *PO* ^{i} *j* також не може існувати. Відсутність реально існуючого *USP* ^{i} *ptr*[*j*] на нього виступає вказівкою на знищення для менеджера системи безпеки.

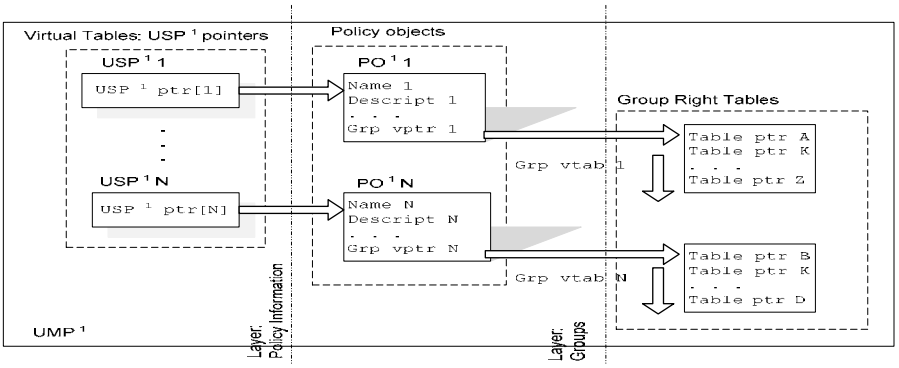

Рис.1. Загальна структура UMP

Згідно даного рисунка належність до даної групи прав визначається існуванням правила:

1) Зауважимо, що комунікація *AMA* можлива тільки з брокером авторизації.
2) Брокер авторизації (autorization brokers, *AB*) є програмним об'єктом, який створюється системою автоматично у відповідь на запити аутентифікації від програмних сервісів. Зауважимо, що визначення категоризації *AB* дає можливість під'єднувати сервіс до існуючого брокера і не створювати у даній *IS* новий.
3) Основним призначенням *AB* є вирішення завдання авторизації. Ц е реалізується через комунікацію брокера з сервісом (чи сервісами) з одного боку і комунікацію з *UMP* з іншого. Загальна структура *AB* наведена на рис.2.

Брокери авторизації повинні існувати двох типів:

1) Прості брокери (Simple autorisation broker, *SAB*)
2) Прозорі брокери (Transparency autorisation broker, *TAB*).

TAB інкапсулює *SBA* і додає до нього лише так званий Transparency layer (рис.2), який необхідний для вирішення завдань синхронізації.

Один *AB* може обслуговувати декілька сервісів. При цьому мережеві сервіси обслуговуються виключно *TAB*. Це зумовлено тим, що *TBA* може негайно чи відкладено запустити процес синхронізацію активного *UMP*-об'єкта стосовно усіх машин мережі. При цьому для кожного *UMP*, задіяного в процесі синхронізації, створюється власний об'єкт, який заноситься в пул *UMP*-об'єктів (рис.3). Це є свідченням того, що об'єкти

засинхронізовані. Цей об’єкт створюється лише на машині, на якій активний об’єкт володіє іншим *UMP*.

Пул *UMP*-об’єктів дає змогу пришвидшити будь-які операції аутентифікації доступу сервісів до мережі.

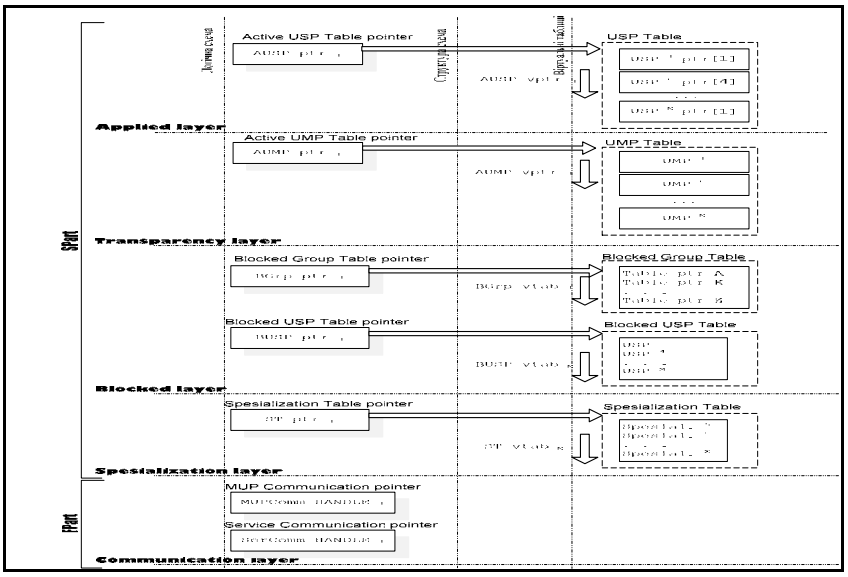

Рис.2 Структурно-логічна схема брокера авторизації

Існування прикладних об’єктів (на рис.3. позначені як safety objects, *SO*) обумовлює утворення так званого пулу *MUP*-об’єктів (*MUP Pool*). Мінімальна кількість *MUP*-об’єктів в пулі визначається кількістю об’єктів, які вимагають авторизації. Окрім того в ньому можуть існувати *MUP* різноманітних мережевих чи віддалених сервісів. Тобто розмір пулу не менший за кількість прикладних об’єктів.

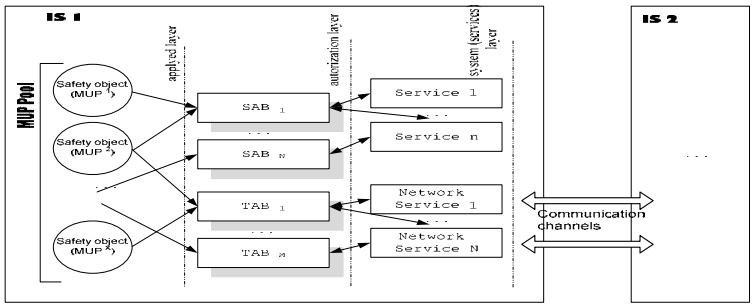

Рис.3. Схема організації доступу в всередині інформаційної системи та в мережевих сеансах

Доступ *SO* до сервісів здійснюється через прошарок *AB*. Це дає можливість строго відділити питання надання доступу від прикладних завдань, для яких потрібна комунікація із сервісами. Більше того, запровадження на Communication Layer всередині брокера будь-яких тунельних чи канальних технологій дає змогу зробити брокерами

прозорими. В цьому випадку брокери можуть ще й відігравати роль безпечних і контрольних інтерфейсів обміну даними між *SO* з одного боку та сервісами з іншого.

Як показано на рисунку доступ до внутрішніх сервісів *IS* забезпечується *SAB*. У випадку доступу до мережевих сервісів, тобто там, де інформація про *MUP* потрібна надалі, треба використовувати *TAB*.

З іншого боку, якщо запитуваний сервіс потребує результатів роботи іншого, то для отримання до доступу до останнього, можливо, також буде потрібний *MUP*. В цьому випадку також необхідно використовувати *TAB*.

Проаналізувавши структуру об'єктних брокерів організації доступу, зроблено висновок що основними перевагами *UMP* є:

1) Можливість існування різних policy для однієї аутентифікації;
2) Автоматичне оновлення *UMP* (реалізовується в результаті синхронізації між різними *AMA*);
3) Можливість для одного *UMA* запускати сервіс (чи сервіси) під різними рахунками (accounts, які визначають права доступу).
4) основними перевагами *AB* є:
5) Відділення логістики авторизації від прикладного рівня;
6) Уніфікація логістики авторизації для різних локальних сервісів і для віддалених сервісів;
7) Можливість синхронізації policy для одного користувача, які створені на різних машинах;
8) Поява можливості організовувати доступ до різних сервісів з різними правами доступу;

1. *Таненбаум. Современные операционные системы. М.: Питер, 2002, 1037с.*
2. *М.Русинович, Д.Соломон. Внутренее устройство Microsoft Windows: Windows Server 2003, Windows XP, Windows 2000. М.: Питер, 2005, 992с.*
3. *Структура об'єктних брокерів організації доступу . Рашкевич, Ю.; Пелешко, Д.;Кустра, Н.; Пасєка, М.*