*УДК 681.3*
**М.Карпінський[1], докт.техн.наук; А.Фурманюк[2]**
[1]*Університет в Бєльску-Бялей, Республіка Польща*
[2]*Тернопільська академія народного господарства*

# АУТЕНТИФІКАЦІЯ У СУБД ТА ЗБЕРЕЖЕННЯ ТАЄМНИЦІ ПАРОЛІВ

*В статті розглядаються поширені методи аутентифікації користувачів різних СУБД, аналізуються їх переваги та недоліки, даються рекомендації із захисту від розкриття паролів доступу до баз даних. Показано шляхи підвищення ефективності аутентифікації за допомогою допоміжних апаратних засобів, наприклад електронного USB-ключа Token Web Sign On фірми Aladdin.*

**Постановка задачі у загальному вигляді та її зв'язок із практичними завданнями.** В сучасних умовах будь-яка господарська діяльність пов'язана з оперуванням великими обсягами інформації, яке виконується широким колом осіб. Захист даних від несанкціонованого доступу є одним із пріоритетних завдань при проектуванні сучасної інформаційної системи. Наслідком збільшення останнім часом значення інформації стали високі вимоги до конфіденційності даних. Системи управління базами даних (СУБД), а особливо реляційні СУБД, стали домінуючим інструментом у цій сфері. Забезпечення інформаційної безпеки СУБД має на меті вибір конкретного засобу забезпечення необхідного рівня безпеки організації в цілому.

Аутентифікація користувачів в СУБД є ключовим моментом захисту конфіденційних даних, що зберігаються в них, від несанкціонованого доступу. Для успішного захисту СУБД необхідним є збереження у таємниці інформації про імена користувачів та особливо паролі доступу до бази даних. Крім того, строга аутентифікація дозволяє захиститися від деяких інших загроз безпеці, наприклад: підміна IP, невірна маршрутизація і підміна DNS.

**Аналіз останніх досліджень та публікацій.** Захист СУБД не обмежується лише внутрішніми захисними механізмами її самої. Необхідно також захистити й операційну систему (ОС), під управлінням якої працює база даних. Тобто захист інформації в системах управління базами даних – це довгий і непростий процес, що полягає не лише в придбанні і настроюванні спеціалізованих засобів захисту, але й у виявленні й усуненні всіх загроз для інформації у СУБД. Без цього можна звести нанівець всю ефективність прийнятих захисних дій. Саме тому останні версії СУБД SQL Server компанії Microsoft та інші високорівневі СУБД викликають інтерес через делеговану в них можливість аутентифікації користувачів, використовуючи вбудовані засоби операційної системи Windows. Наскільки такий метод є ефективним у розв'язанні задачі доступу до конфіденційної інформації, йтиметься далі.

**Формування цілей статті.** Метою даної роботи є дослідження ефективності методів аутентифікації користувачів популярних у наш час реляційних СУБД та вироблення рекомендацій зі збереження таємниці паролів як головного моменту недопущення розкриття конфіденційної інформації.

**Виклад основного матеріалу.** Зазвичай в СУБД для ідентифікації і перевірки дійсності користувачів використовуються або відповідні механізми операційної системи, або SQL-оператор CONNECT. Наприклад, у випадку СУБД Oracle оператор CONNECT має наступний вигляд:

```
CONNECT login[/password] [@dbms];
```

де login і password – це ім'я користувача і пароль, а dbms – відповідно ім'я бази даних.

Так чи інакше, у момент початку сеансу роботи із сервером баз даних, користувач ідентифікується своїм ім'ям, а засобом аутентифікації слугує пароль. Деталі цього процесу визначаються реалізацією клієнтської частини програми.

Фактичні імена користувачів (ідентифікатори) і паролі зберігаються в одній із таблиць бази даних на сервері. Розглянемо методи аутентифікації у деяких популярних СУБД:

**IBM DB2.** Ця СУБД наділена низкою методів аутентифікації, які логічно можна згрупувати у чотири види: серверна, клієнтська, DCI та Kerberos [1].

1. Серверна аутентифікація – механізм захисту за замовчуванням, визначає, що ідентифікація має відбутися на сервері, використовуючи операційну систему сервера. Якщо ім'я користувача (ідентифікатор) і пароль визначені протягом приєднання до СУБД, DB2 викликає функцію операційної системи, щоб перевірити їх правильність. В середовищі Windows комбінація ідентифікатора користувача і пароля часто згадується як обліковий запис користувача. Опцією серверної аутентифікації є SERVER_ENCRYPT, яка відрізняється від заданої за замовчуванням тим, що пароль приходить від клієнта на сервер у зашифрованому вигляді. DB2 використовує DES-криптування паролю і алгоритм Діффі-Хелмана, щоб генерувати ключ для алгоритму криптування під час з'єднання. Цю підтримку забезпечує інструментарій RSA BSAFE.
2. При клієнтській аутентифікації, що має місце на стороні клієнта, використовуються механізми захисту операційної системи. При цьому вважається, що сервер бази даних „довіряє" клієнту, що неможливо при використанні незахищених операційних систем Windows 95, 98.
3. DCE. Багато адміністраторів обирають цей метод, тому що він забезпечує централізоване адміністрування користувачів і паролів та не допускає передачі ідентифікаторі і паролів у „чистому", тобто незашифрованому вигляді. DB2 клієнт, який зареєстрований в DCE може отримати зашифрований „квиток", який використовується як доказ ідентифікації у DB2.
4. Kerberos [2] – це сервер захищених імен користувачів і паролів. Перевага Kerberos полягає у тому, що він забезпечує централізований захист всіх даних і ресурсів у мережі. Доступ до бази даних, вхід у систему, управління ресурсами та інші особливості захисту централізовано розміщені на „довіреному" сервері Kerberos. Об'єднаний із системою розподілу публічних ключів (public keys) Kerberos є потужним інструментом у побудові захищених розподілених комп'ютерних мереж. Аутентифікація за допомогою Kerberos підтримується у більшості сучасних СУБД.

**MySQL.** Підтримує наступні методи аутентифікації: .rhost з RSA, просто RSA, одноразовий пароль s/key і Kerberos. RSA-аутентифікація грунтується на неможливості отримання приватного ключа з публічного. Публічний ключ зберігається на сервері у користувача в файлі $HOME/.ssh/authorized_keys. Приватний ключ зберігається тільки на локальній машині користувача або іншому таємному носії. Приватний ключ RSA може бути захищений, в свою чергу, парольною фразою (passphrase). Нею може бути будь-який рядок, він хешується з допомогою MD5 для створення ключа шифрування для алгоритму 3DES, який використовується для шифрації файла приватного ключа.

При роботі із MySQL версії 3.22.32 була виявлена вразливість функції password() [3]. MySQL зберігає в себе паролі, що не допускають їх розшифрування (підбір можливий лише з допомогою brute-force). У відповідь на запит користувача сервер випадковим чином генерує рядок (random string). На основі дайджеста (хеша) цього рядка і хеша пароля клієнт обчислює "check" рядок, яка потім відсилається на сервер, який також обчислює "check" рядок і порівнює її з отриманим. Хеш-функція, що використовувалась в MySQL версії 3.22.32, повертала рядок розміром 64 біт, не дивлячись на те, що функція, яка генерує random string, - 40 біт. Для проходження успішної аутентифікації потрібен не сам пароль, а його хеш, який вдалося визначити дослідникам вразливості. У версії MySQL 3.23.2 [4] вразливість була усунена; почалось використання алгоритму MD5, що видає 128-бітні дайджести.

**Microsoft SQL Server.** Підтримує два режими аутентифікації – інтегрований (Windows Authentication Mode) та змішаний (Mixed Mode). При інтегрованому режимі для користувача задається один обліковий запис в операційній системі як користувача домену, а SQL-сервер ідентифікує користувача за його даними у цьому обліковому записі. В цьому випадку користувач задає тільки одне своє ім'я і один пароль. У випадку змішаного режиму частина користувачів може бути приєднана до сервера з використанням інтегрованого режиму, а іншій частині потрібно буде додатково ввести свої унікальні імена користувачів SQL Server і свої унікальні паролі. Алгоритм перевірки аутентифікації користувача в MS SQL Server 7.0 наведений на рис. 1.

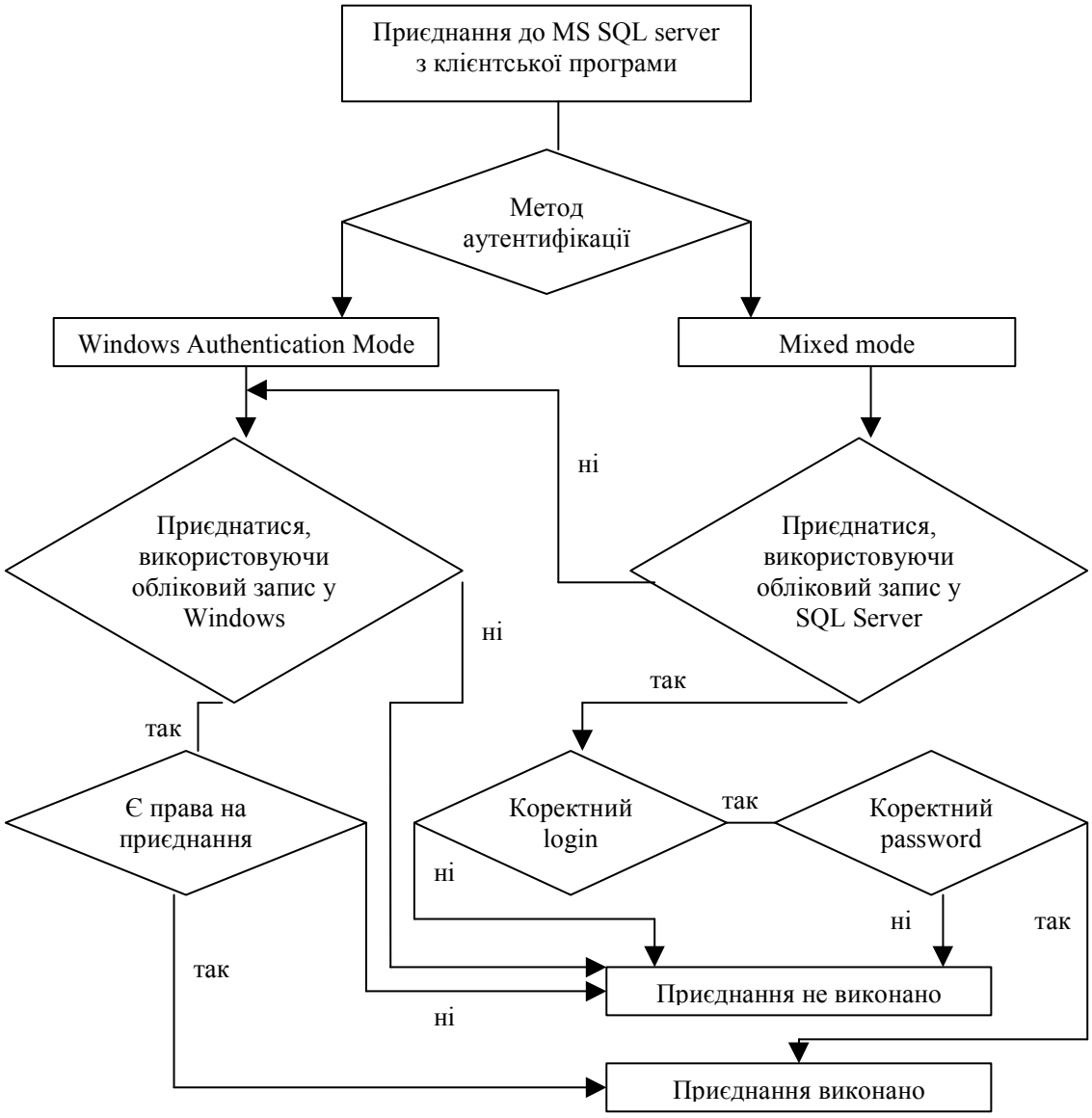


Рис. 1. Аутентифікація у MS SQL Server 7.0

У СУБД **Oracle** в доповнення до імені користувача і його пароля в операційній системі завжди використовуються ім'я і пароль для роботи із сервером баз даних.

Отже, після вищезазначеного стає зрозумілим, що для забезпечення ефективної аутентифікації потрібно зберігати паролі на сервері в зашифрованому вигляді і не передавати їх у вигляді „відкритого" тексту по незахищених каналах зв'язку, наприклад через Інтернет. Безпечні web-з'єднання встановлюються з допомогою протоколу SSL. Цей протокол використовує відкрите шифрування ключів для встановлення з'єднання, при якому тільки сторони, що беруть участь в обміні, володіють ключем розшифрування даних, що проходять через з'єднання. Така організація роботи виключає можливість перегляду інформації третіми особами. Протокол SSL містить

засоби, які дозволяють виявляти будь-які зміни, втрати і повторення даних. В протоколі SSL також використовуються алгоритми для проведення ідентифікації з допомогою стандарту X509 [5].

Незважаючи на зручність та різноманітність методів аутентифікації, вони мають деякі недоліки. Наприклад, MS SQL Server не забезпечує можливості блокування облікового запису користувача бази даних у випадку серії невдалих спроб аутентифікації. Це дозволяє зловмиснику здійснювати різні атаки на систему ідентифікації/аутентифікації, намагаючись підібрати імена користувачів, зареєстрованих в СУБД, та їх паролі. Другий приклад вразливості - неможливість перейменування облікового запису системного адміністратора бази даних (sa), що також дозволяє здійснювати зловмиснику спроби підбору пароля адміністратора СУБД. Ця вразливість має місце не лише в базах даних, а і в операційних системах і прикладному програмному забезпеченні. Іншим недоліком практично всіх СУБД є відсутність перевірки ефективності обраного користувачем пароля. Нерідко у користувача зовсім відсутній пароль. До чого це може призвести, говорити зайве. Також користувачі можуть місяцями не користуватися базою даних, але, будучи один раз в неї внесеними, вони вважаються повноправними її користувачами. В СУБД і багатьох ОС відсутні механізми контролю облікових записів, які не використовувались протягом заданого проміжку часу.

**Висновки і перспективи подальшого розвитку.** Правильний вибір методу аутентифікації та ефективне використання криптоалгоритмів ще не означає, що ваші дані у цілковитій безпеці. Паролі доступу до СУБД повинні зберігатися в зашифрованому вигляді, передаватися по захищених каналах зв'язку, а машина клієнта має бути захищена від несанкціонованого доступу. Є надія, що описані в статті недоліки будуть виправлені у наступних версіях СУБД та операційних системах, під управлінням яких СУБД працюють. Проте є можливість підвищити ефективність аутентифікації зараз, використовуючи допоміжні апаратні засоби. Прикладом може бути електронний USB-ключ eToken Web Sign On фірми Aladdin [6]. Цей пристрій хоч і не містить вбудованих засобів криптографічного захисту, проте зберігає у своїй пам'яті ідентифікатори і паролі до різноманітних html-форм, що є дуже поширеними, коли доступ до бази даних відбувається через web. Використання ключа унеможливлює перехоплення ідентифікатора і пароля так званими „клавіатурними шпигунами", що можуть бути присутніми на незахищених робочих станціях, оскільки html-форми заповнюються автоматично без вводу даних з клавіатури. Ключ також містить у собі генератор випадкових паролів.

*In article popular authentication methods of users various DBMS are considered, their merits and demerits are analyzed, recommendations on protection against disclosing passwords of access to databases are given. The ways to increase of authentication effective by the subsidiary hardware, for electronics USB-key Token Web Sign On the Aladdin firm production.*

### Література

1. Zikopoulos P. The database security blanket, http://www.governmentsecurity.org.
2. Бернет С., Пэйн С. Криптография. Официальное руководство RSA Security. – М.: Бином, 2002. – 374 с.
3. Нестойкость password() в MySQL, http://www.void.ru.
4. http://www.mysql.com. Open source database MySQL.
5. Дюбуа П. Применение MySQL и Perl в Web-приложениях. – М.:Вильямс, 2002. – 371 с.
6. http://www.aladdin.ru. Aladdin Software Security R.D.

*Одержано 05.01.2005 р.*