STUDIA INFORMATICA 2010
Volume 31 Number 2B (90)

Katarzyna HARĘŻLAK
Politechnika Śląska, Instytut Informatyki

BEZPIECZEŃSTWO MOBILNYCH BAZ DANYCH

Streszczenie. W artykule przeprowadzono analizę mechanizmów gwarantujących bezpieczeństwo w systemach wykorzystujących mobilne bazy danych. Analiza ta ma stanowić wskazówki oraz wytyczne dla twórców i administratorów takich systemów rozproszonych. Badania prowadzone były w środowisku zbudowanym z narzędzi firmy Microsoft – IIS, SQL Server 2008 oraz SQL Server Mobile.

Słowa kluczowe: mobilne bazy danych, bezpieczeństwo, replikacja danych

MOBILE DATABASES SECURITY

Summary. The mobile distributed system architecture analysis concerning mechanisms guaranteeing the data safety was presented in the paper. This analysis can constitute a set of guidelines for developers and administrators of such systems. The research environment was created with usage of Microsoft Tools – IIS, SQL Server 2008 and SQL Server Mobile.

Keywords: mobile databases, data security, data replication

1. Wstęp

Obecnie, w czasach powszechnego zastosowania urządzeń mobilnych wspomagających działania wielu firm, wzrastają wymagania w zakresie dostępności informacji zawartych w bazach danych. Oczekuje się ich obecności również na urządzeniach przenośnych, jakimi są palmtopy czy telefony komórkowe, ponieważ w wielu przypadkach posiadanie informacji lub jej brak decyduje o powodzeniu firmy w jej sztandarowych działaniach. Dlatego równie ważnym elementem, jak pozyskiwanie i przetwarzanie informacji, jest jej zabezpieczenie. Jednak temat bezpieczeństwa w zakresie tworzenia oraz rozwoju aplikacji mobilnych jest często tematem pomijanym lub spychanym na margines [9, 11].

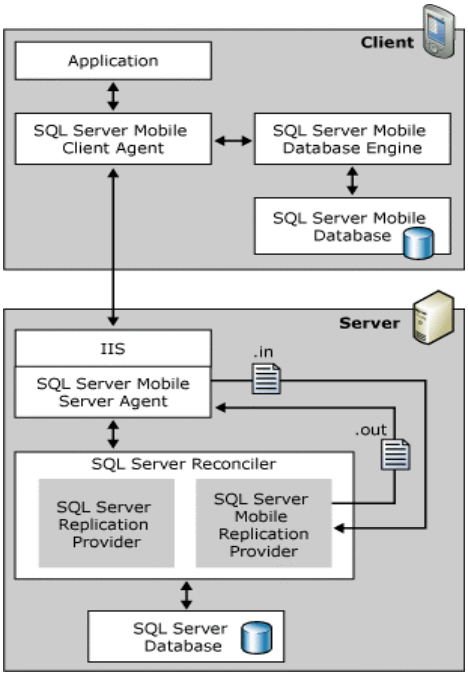
Można znaleźć wiele prac dotyczących przetwarzania informacji w mobilnych bazach danych [1, 4, 6, 7, 12, 12], trudno jednak spotkać takie, które odnoszą się do zabezpieczania procesu replikacji danych z centralnej bazy danych na urządzenia przenośne. W pracach [2, 5] poruszono, co prawda, temat bezpieczeństwa mobilnych baz danych, aczkolwiek głównie w zakresie ochrony dostępu do urządzenia przenośnego oraz przesyłu informacji pomiędzy elementami systemu mobilnego. W niniejszej pracy uwagę skupiono na bezpieczeństwie całego procesu replikacji danych do mobilnej bazy danych. Jest on ściśle związany ze stosowanym środowiskiem działania, tym niemniej dla każdego z nich można wyróżnić kilka podstawowych elementów.

1. **Konfiguracja procesu replikacji w centralnej bazie danych.** Bazodanowy system informatyczny posiadający moduł dedykowany urządzeniu mobilnemu zawiera w swojej architekturze centralną bazę danych i jej replikę na urządzeniu przenośnym. Replika ta powstaje przez powielenie danych z centralnej bazy danych w określonym momencie czasu i musi być okresowo z nią synchronizowana. W wielu przypadkach dane w centralnej bazie danych są danymi poufnymi, zatem zarówno skonfigurowanie, uruchomianie procesu replikacji danych oraz późniejszej ich synchronizacji powinno wymagać odpowiednich praw dostępu, uniemożliwiających nielegalny odczyt danych na urządzenie mobilne oraz ich modyfikacji w trakcie synchronizacji.
2. **Ochrona komunikacji modułów systemu**. Połączenie modułów systemu zawierającego część mobilną może być realizowane różnymi sposobami. Pierwszy z nich, najprostszy, zakłada wykorzystanie portu USB do skomunikowania urządzenia mobilnego z komputerem stacjonarnym. W takim przypadku ryzyko przechwycenia danych podczas transmisji nie jest znaczące. Większe niebezpieczeństwo niesie ze sobą komunikacja bezprzewodowa, co wymusza odpowiednią jej konfigurację i szyfrowanie połączeń.
3. **Zabezpieczenie dostępu do urządzenia mobilnego**, które ze swej natury podatne jest na łatwy dostęp przez nieupoważnione osoby, narzuca konieczność stosowania mechanizmów identyfikacji użytkownika przy próbie skorzystania z zasobów urządzenia.
4. **Autoryzacja i uwierzytelnianie użytkowników aplikacji dedykowanej urządzeniu mobilnemu.** Na urządzeniu przenośnym z baz danych korzysta się głównie za pośrednictwem dedykowanej aplikacji, zatem uzyskanie dostępu do jej funkcji otwiera drogę do odczytu i modyfikacji znajdujących się tam rekordów. Do twórcy takiej aplikacji należy przygotowanie odpowiedniej polityki i mechanizmów zabezpieczających przed możliwością nieuprawnionego uruchomienia realizowanych przez nią zadań.
5. **Kontrola dostępu do mobilnej bazy danych.** Serwery baz danych, udostępniające możliwość tworzenia swoich odpowiedników na urządzenia przenośne, gwarantują również zestaw narzędzi pozwalających zarządzać tymi danymi w środowisku mobilnym. Wynika z tego, że odczyt rekordów z mobilnej wersji bazy danych może zostać rozszerzony poza

granice aplikacji użytkownika. Należy więc przeanalizować sposoby ochrony mobilnej baz danych, które zazwyczaj nie posiadają tak rozbudowanych mechanizmów identyfikacji i autoryzacji, jak serwery stacjonarne. Ten sam problem dotyczy reguł kontroli poprawności wprowadzanych i modyfikowanych danych oraz utraty bazy danych w wyniku awarii.

2. System mobilny w środowisku zbudowanym w oparciu o narzędzia firmy Microsoft

Ochrona środowiska mobilnego systemu wykorzystującego bazy danych wymaga podjęcia działań na wszystkich, omówionych w poprzednim rozdziale, poziomach. W artykule analizę takich mechanizmów przeprowadzono dla stanowiska badawczego składającego się z: komputera stacjonarnego z systemem zarządzania baz danych firmy Microsoft, SQL Server 2008 oraz urządzenia przenośnego typu Pocket PC z systemem operacyjnym Windows Mobile 5.0 wyposażonego w bazę danych systemu SQL Server Mobile (rys.1) [10].

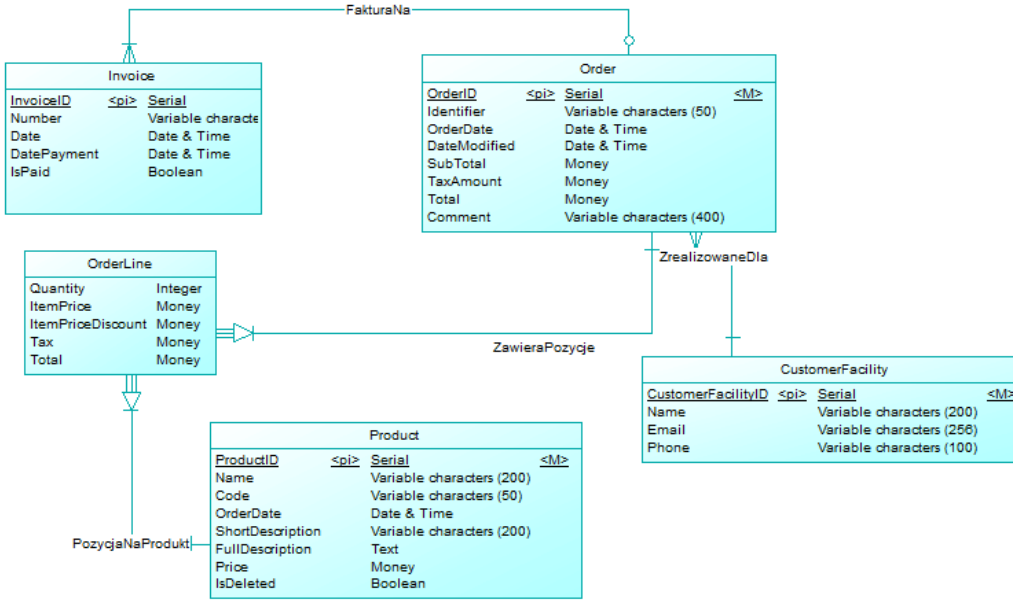


Rys. 1. Architektura mobilnego systemu rozproszonego w środowisku Microsoft
Fig. 1. Architecture of mobile distributed system in Microsoft environment

Współdziałanie baz danych znajdujących się na obu serwerach opiera się na współpracy wielu elementów takiego środowiska, z których każdy powinien podlegać właściwym zasadom ochrony.

W celu przeprowadzenia badań zaprojektowano prostą strukturę bazy danych obsługującą system sprzedaży mobilnej (rys. 2). Wchodząca w jej skład tabela *Order* rejestruje zamówie-

nia klienta, opisane odpowiednim nagłówkiem zawierającym miedzy innymi datę utworzenia, modyfikacji, wartość zamówienia netto, wartość podatku oraz wartość zamówienia brutto. Zamówienie posiada wiele pozycji (tabela *OrderLine*), każda z nich jest sporządzona dla konkretnego produktu (tabela *Product*) z uwzględnieniem ilości, ceny za pojedynczy produkt oraz za całą pozycję. Zamówienia realizowane są dla konkretnej placówki klienta (tabela *CustomFacility*) i mogą mieć kilka powiązanych faktur (tabela *Invoice*) .



Rys. 2. Diagram ER wykorzystanej w badaniach bazy danych
Fig. 2. ER diagram from used in research database

2.1. Centralna baza danych

Pierwszym etapem, który musi pokonać użytkownik chcący zarządzać tworzeniem repliki bazy danych na urządzeniu przenośnym, jest zalogowanie się do serwera, w którym znajduje się centralna baza danych będąca źródłem danych dla replikacji. Dostęp do serwera centralnego w omawianym środowisku może być realizowany na dwa sposoby.

1. **Uwierzytelnianie Windows**, w którym wykorzystywany jest protokół *Kerberos* przydzielający uprawnienia do zasobów na podstawie hasła zweryfikowanego w trakcie logowania użytkownika do systemu operacyjnego. Taki rodzaj dostępu do systemu SQL Server musi być poprzedzony rejestracją użytkownika w serwerze. Dla przykładu, jeżeli w domenie firmy istnieje konto użytkownika *Admin*, to dostęp dla niego uzyskać można za pomocą polecenia

   ```
   create login [firma\admin] from Windows,
   ```

   gdzie *firma\admin* jest kontem *admin* należacym do domeny *firma*.

2. Drugi sposób, zwany **uwierzytelnianiem** systemu **SQL Server**, wymaga podania nazwy konta oraz hasła użytkownika, które przechowywane są w tabelach systemowych serwera. W trakcie tworzenia takiego konta poleceniem:

```
create login sprzedaz with password = 'X22DS45',
```

hasło poddawane jest działaniu funkcji mieszającej, a jej wynik zapisywany jest w serwerze – nie ma możliwości jego odtworzenia, nawet dla właściciela konta. Sytuację tę prześledzić można analizując zamieszczony dalej wydruk przedstawiający wynik zapytania do odpowiednich tabel systemowych. Rekordy, które mają w polu *password* wartość NULL, są kontami systemu operacyjnego zarejestrowanymi w serwerze baz danych, pozostałe rekordy to konta utworzone w trybie uwierzytelniania SQL Server.

```
select name,
       master.dbo.fn_varbintohexstr(cast(password as varbinary(255)))
       from sys.server_principals

   name             fn_varbintohexstr(cast(password as varbinary(255)))
   ------------------------------------------------
   sa               0x01004086ceb6d8277477b39b7130d923f399c6fd3c6bd46a0365
   mobilny          0x010090788ab2f2c35ed4cf9ca81bdae2482735f81d4c31e98803
   firma\agent      NULL
   sprzedaz         0x01004086ceb6d8277477b39b7130d923f399c6fd3c6bd46a0365
```

2.2. Bezpieczeństwo procesu replikacji danych

Jak wspomniano wcześniej, utworzenie bazy danych na urządzeniu przenośnym zazwyczaj wiąże się z procesem jej replikacji z serwera centralnego. Ochrona takiego procesu w narzędziach firmy Microsoft zorganizowana została na wielu poziomach.

Pierwszy z nich dotyczy konfiguracji środowiska rozproszonego i włączenia w nie określonego serwera w roli *wydawcy* (*ang. publisher*), aby mógł udostępniać swoje dane.

```
   exec sp_adddistpublisher
   @publisher='firma',           'nazwa serwera, który staje się wydawcą
   @distribution_db='distribution','systemowa baza danych przechowująca dane
                                  o środowisku rozproszonym
@security_mode = 1; 'sposób uwierzytelniania procesów replikujących dane – w tym
                    przypadku uwierzytelnianie Windows
```

Realizacją takiego zadania może zająć się wyłącznie taki użytkownik, któremu nadano uprawnienia administracyjne, co realizowane jest poprzez przypisanie go do roli *sysadmin* za pomocą procedury:

```
sp_addsrvrolemember [sprzedaz], sysadmin
```

Takie same uprawnienia trzeba posiadać, by nadać serwerowi prawa *subskrybenta* (ang. *subscriber)* publikowanych przez wydawcę danych. Tylko administrator serwera może zarejestrować go w tej roli w środowisku rozproszonym.

Drugi z **poziomów** zabezpieczeń obejmuje prawa, które pozwalają użytkownikowi zmieniać właściwości określonej bazy danych tak, by można było publikować jej rekordy. Możliwość przydzielania takich przywilejów zarezerwowana jest również tylko dla administratora serwera ustanowionego jako wydawcę.

Ostatni **poziom** zabezpieczania konfiguracji środowiska replikacji danych dotyczy tabel. Każdy z obiektów, którego rekordy mają być powielane do innego węzła, włączany jest w *publikacje (*ang. *publications)* – definiujące zakres udostępnianych danych i metodę ich replikacji. Natomiast szczegóły określające jej miejsce zawarte są w obiektach zwanych *subskrypcjami (*ang. *subscryption)*. W obu przypadkach, w celu zarządzania takimi obiektami, należy uzyskać uprawnienie albo administratora serwera, albo właściciela baz danych (rola **db_owner**):

```
sp_addrolemember db_owner, [firma\admin],
```

w konsekwencji czego uzyskuje się prawo do wykorzytywania następujących procedur:

- tworzenia publikacji – procedura *sp_addmergepublication* – posiada wiele parametrów konfiguracyjnych, część z nich dotyczy także ochrony dostępu do danych:

```
exec sp_addmergepublication
@publication = N'mobilna',
@allow_anonymous = N'True' 'wartość True dopuszcza anonimowe subskrypcje, co
                        jest niezbędne przy replikacji do mobilnych baz danych.
@allow_subscription_copy = N'false' 'zezwala bądź nie na kopiowanie bazy danych,
                                    która zawiera subskrypcję
@allow_subscriber_initiated_snapshot = N'false', 'określa, czy subskrybent sam
                                                  może inicjować replikę
@allow_web_synchronization = N'true'    'wartość TRUE, umożliwia synchronizację
                                    z użyciem protokołu HTTPS, niezbędne
                                    w przypadku mobilnych baz danych
```

- dołączania tabeli do publikacji – *sp_addmergearticle* – zaprezentowanej dla tabeli *Order* (rys. 2):

```
exec sp_addmergearticle @publication = N'mobilna',
@article = N'Order',
@source_owner = N'dbo',
@source_object = N'Order',
@type = N'table',
@destination_owner = N'dbo',
```

Ponadto, utworzona publikacja może być udostępniana różnym użytkownikom reprezentowanym przez konta serwerowe, a udzielenie im praw do korzystania z niej może odbywać się z pomocą wbudowanej procedury:

```
exec sp_grant_publication_access
@publication = N'mobilna,
@login = N'sprzedawca'.
```

gdzie *mobilna* reprezentuje nazwę publikacji, a *sprzedawca* jest kontem, które będzie mogło replikować dane z tej publikacji.

Dodatkowym zabezpieczeniem dla ochrony danych jest wprowadzenie do publikacji filtrów ograniczających zestaw publikowanych danych. Przedstawiony dalej przykład zawęża zestaw zamówień do tych, które dotyczą konkretnego, zalogowanego sprzedawcy (`@user_id`).

```
SELECT <published_columns> FROM [dbo].[Order] WHERE EmployeeID = @user_id
```

2.3. Procesy tworzące środowisko replikacji danych

Omówione wcześniej elementy, tworzące bazodanowe środowisko rozproszone, stanowią jego część statyczną, określając, skąd dane będą pobierane i jakie jest ich miejsce przeznaczenia. Jednak sama operacja powielania i synchronizacji danych z różnych węzłów sieci komputerowej realizowana jest przez zestaw procesów zwanych *agentami*. Każdy z nich odpowiedzialny jest za realizację określonych funkcji, zgodnie z konfiguracją zawartą w definicji *publikacji* lub *subskrypcji*, i powoływany jest do życia w ramach określonego konta systemu operacyjnego, stanowiącego kontekst jego pracy. Zaleca się więc, by każdy z nich związany był z innym kontem systemowym, któremu przydzielono tylko niezbędne uprawnienia, stosowne do jego zadań.

Duże znaczenie ma w tym miejscu konfiguracja praw użytkowników zarówno na poziomie systemu operacyjnego, jak i w środowisku systemu SQL Server. Pierwsza grupa uprawnień związana jest z zakresem działania agenta poza serwerem bazy danych. Obejmuje ona dostęp do zasobów systemu operacyjnego lub domeny, w jakiej zarejestrowany został komputer z zainstalowanym systemem SQL Server. Druga grupa uprawnień zabezpiecza możliwość lub jej brak w odniesieniu do działań, które chce wykonać agent wewnątrz serwera bazy danych.

W przypadku replikacji danych na urządzenia przenośne rozważania dotyczą dwóch typów agentów: *Snapshot*, tworzącego migawkę początkową powielanych danych, oraz *Merge*, który odpowiedzialny jest za okresową ich synchronizację.

Uruchamianie procesów z poziomu serwera bazy danych

W środowisku serwera omawiani agenci widziani są jako zadania administracyjne, które uruchamiane są przez proces zwany *SQL Server Agent*, zgodnie z zadanym harmonogramem pracy. Jeśli zalogowanym do serwera użytkownikiem nie jest konto administracyjne, to uprawnienie do wykonania takiego zadania można nadać mu przypisując go do jednej z podanych dalej bazodanowych ról:

- **SQLAgentUserRole** – zawiera uprawnia do tworzenia, przeglądania i śledzenia własnych zadań,
- **SQLAgentReaderRole** – posiada własności roli **SQLAgentUserRole,** rozszerzając je o te same prawa w odniesieniu do dowolnego zdania,

- **SQLAgentOperatorRole** – udostępnia, poza wcześniej wymienionymi uprawnieniami, możliwość zarządzania wszystkimi zadaniami i skojarzonymi z nimi obiektami administracyjnymi.

Dla przykładu, przydzielenie uprawnień użytkownikowi *sprzedaz* do tworzenia zadań administracyjnych na serwerze centralnym prezentuje zamieszczony dalej kod:

```
sp_addrolemember SQLAgentUserRole, sprzedaz
```

Dla kont, którym nie przydzielono powyższych przywilejów, zadania uruchamiające agentów replikacji nie są na serwerze dostępne.

Uruchamianie procesów z poziomu systemu operacyjnego

Każdy z wymienionych agentów może być uruchamiany również z poziomu systemu operacyjnego i z tego punktu widziany jest jako zwykły program wykonywalny. W pierwszym kroku należy więc zabezpieczyć dostęp do jego lokalizacji użytkownikom do tego nieupoważnionym. Przypadkowe uruchomienie takiego programu przez niepowołaną osobę nie musi skutkować zainicjowaniem danego kroku replikacji danych. Uruchomiony z linii poleceń agent przejmuje uprawnienia użytkownika, który zainicjował jego pracę, na co wskazują parametry wywołania (`-subscriberSecurityMode 1, -distributorSecurityMode 1, -publisherSecurityMode 1`).

Agent typu *Snapshot*

```
snapshot.exe -publication mobilna -publisher firma -publisherDB baza_zamowien
-replicationtype 1 -distributor kmhar  -distributorSecurityMode 1
-publisherSecurityMode 1
```

Agent typu *Merge*

```
replmerg.exe -publisher firma -publisherdb baza_zamowien -subscriber fir-
ma_mobilna -subscriberDB sprzedaz -distributor kmhar -publication mobilna
–subscriptiontype 1 -subscriberSecurityMode 1 -distributorSecurityMode 1
-publisherSecurityMode 1
```

Zatem jeśli użytkownik uruchamiający powyższe polecenia nie ma praw dostępu do serwera lub do wykonywania operacji, które zapisane są w scenariuszu agenta, mechanizmy zabezpieczeń systemu SQL Server zablokują takie próby.

Zmiana dowolnego parametru z grupy *`SecurityMode` wymusza uwierzytelnianie agenta w trybie SQL Server, co pociąga za sobą konieczność podania wartości parametrów z grupy `DistributorLogin`, `DistributorPassword`, `PublisherLogin`, `PublisherPassword`, `SubscriberLogin`, `SubscriberPassword`. Ale wtedy trzeba znać nazwę odpowiedniego konta znajdującego się na serwerze oraz jego hasło.

2.4. Komunikacja serwera centralnego z mobilną bazą danych

Komunikacja modułów rozproszonego systemu wykorzystującego mobilne bazy danych odbywa się za pośrednictwem Informacyjnych Usług Internetowych ((IIS), Microsoft Information Services). Podstawą tej komunikacji jest *SQL Server Mobile Server Agent*, który w trakcie konfiguracji mobilnego środowiska replikacyjnego musi zostać zarejestrowany na serwerze usług internetowych w odpowiednim katalogu wirtualnym. Na poziomie tego katalogu odbywa się kontrola dostępu do zarejestrowanej usługi, dlatego zaleca się stworzenie osobnego katalogu wirtualnego niezależnie dla każdego urządzenia przenośnego, co pozwala na zróżnicowanie uprawnień dla różnych klientów mobilnych.

Uwierzytelnianie przy dostępie do katalogu wirtualnego serwera WWW może odbywać się na trzy sposoby.

1. Pierwszy z nich zakłada anonimowy dostęp do serwera, bez podawania nazwy konta i hasła, a klient działa wtedy na serwerze z uprawnieniami przygotowanego w tym celu konta systemu operacyjnego. Domyślnie jest to konto o *IUSER_<nazwa_komputera>*. Przyjęcie takiego sposobu uwierzytelnienia nie pozwala więc na rozróżnianie klientów replikujących dane.
2. Druga z możliwości, to zdefiniowanie na serwerze IIS odpowiedniego użytkownika logowania do serwera, ale podczas uwierzytelniania hasło przesyłane jest otwartym, łatwym do przechwycenia tekstem. Dlatego zaleca się używanie tego sposobu uwierzytelniania w połączeniu z zastosowaniem szyfrowanego kanału SSL.
3. Ostatnia z możliwości zakłada korzystanie z uwierzytelniania Windows, które jest najbezpieczniejszym z wymienionych sposobów. Nie można go jednak zastosować w analizowanym, mobilnym środowisku rozproszonym, ponieważ nie jest ono obsługiwane ze względu na brak protokołu *Kerberos* w systemie Windows Mobile.

2.5. Zabezpieczenia danych na poziomie urządzenia mobilnego

Ochrona danych na urządzeniu przenośnym obejmuje swym zakresem cztery obszary: dostęp do urządzenia fizycznego, dostęp do funkcjonującej na nim aplikacji, bezpieczeństwo uruchamianych aplikacji oraz dostęp do mobilnej bazy danych za pomocą narzędzi klienckich serwera.

Ochrona dostępu do aplikacji

System operacyjny Windows Mobile posiada wiele funkcji oraz udogodnień, które mogą być używane do tworzenia strategii bezpieczeństwa do ochrony urządzenia oraz aplikacji, które się na nim znajdują. Jednym z takich mechanizmów jest komponent architektury systemu operacyjnego zwany LASS (Local Authentication SubSystem) [8]. Jest to moduł, który pozwala na zastosowanie zaawansowanych mechanizmów i polityk uwierzytelniania użyt-

kowników urządzenia wyposażonego w mobilną wersję systemu Windows. Zdefiniowanie takiego mechanizmu polega na instalacji modułu zwanego LAP (Local Authentication Plugin), dynamicznie ładowanej biblioteki (DLL), eksportującej określone funkcje, które wywoływane są przy uruchamianiu systemu w celu uwierzytelnienia użytkownika. Instalowany moduł LAP dodatkowo musi posiadać specjalny podpis cyfrowy, który uzyskuje się poprzez przeprowadzany przez firmę Microsoft proces weryfikacji konkretnego modułu. Tworząc moduł LAP można odwoływać się do wielu serwisów udostępnianych przez środowisko Windows Mobile, takich jak np. *CryptoAPI* – warstwy systemu Windows odpowiedzialnej za udostępnianie usług kryptograficznych.

W badaniach metody kryptograficzne zostały użyte w celu zaszyfrowania wszystkich haseł użytkowników. Zaszyfrowane hasło tworzone jest przy użyciu algorytmu SHA1 [3] z ciągu znaków podanych przez użytkownika i wygenerowanego klucza (Salt). Każdorazowo podczas logowania podawany, jako hasło, ciąg jest kodowany tym samym algorytmem z użyciem klucza z bazy danych, co pozwala na weryfikację poprawności wprowadzonego ciągu. Omówione mechanizmy prezentuje zamieszczony dalej fragment kodu.

```
private static string CreateSalt(int size){
    RNGCryptoServiceProvider provider = new RNGCryptoServiceProvider();
    byte[] data = new byte[size];
    provider.GetBytes(data);
    return Convert.ToBase64String(data);}

private static string CreatePasswordHash(string Password, string Salt){
    string hashed = "";
    SHA1 sha1 = new SHA1CryptoServiceProvider();
    byte[] hash =
    sha1.ComputeHash(System.Text.Encoding.UTF8.GetBytes(Password+Salt));

    foreach (byte b in hash)
        hashed += String.Format("{0,2:X2}", b);

return hashed;}
```

Podpisywanie kodu

Innym mechanizmem polityki bezpieczeństwa dotyczącym uruchamiania aplikacji na urządzeniu przenośnym jest podpisywanie kodu. W celu potwierdzenia tego, czy kod jest podpisany, czy sygnatura jest ważna i zgadza się z autoryzowanym certyfikatem zainstalowanym na urządzeniu, Windows Mobile sprawdza każdy moduł należący do grupy dynamicznych bibliotek dołączanych (.dll) lub plików wykonywalnych (.exe), w momencie kiedy są one ładowane na urządzenie. Działania te są bardzo istotne, gdyż uruchamianie procedury synchronizacji danych inicjowane są ze strony aplikacji mobilnej. Jak widać w załączonym poniżej kodzie, może to pociągać za sobą utworzenie nowej mobilnej bazy danych.

```
public void PerformAsyncSynchronization(){
    if (this.isSynchronizing){
        return;
        }
```

```
    this.isSynchronizing = true;
    try{
        //jesli mobilna baza danych nie istenije, tworzy subskrypcję
        if (!File.Exists(Settings.DatabaseFile)){
            replication.AddSubscription(AddOption.CreateDatabase);}
        OnSyncStarted();
        //synchronizacja
        this.asyncResult = replication.BeginSynchronize(
            new AsyncCallback(SyncCompletedCallback),
            null,
            null,
            new OnSynchronization(OnSynchronizationCallback),
            replication);}
    catch (Exception ex){
            OnSyncError(ex);}
}
```

Zabezpieczenia na poziomie mobilnej bazy danych

W mobilnej wersji systemu SQL Server nie ma możliwości tworzenia użytkowników i nadawania im określonych uprawnień do obiektów bazy danych. Zatem wszystkie osoby, które weszły w jej posiadanie, mogą korzystać z niej bez ograniczeń. Istnieją jednak dwa mechanizmy, które pozwalają nam się chronić przed taką sytuacją. Należą do nich: zabezpieczanie hasłem oraz szyfrowanie.

Zabezpieczanie hasłem – SQL Server Mobile pozwala na utworzenie jednego hasła dla całej bazy danych, które wymagane jest następnie przy każdej próbie jej użycia. Zabezpieczanie bazy danych hasłem realizowane jest w trakcie procesu jej tworzenia, z wykorzystaniem następujących metod:

1. Polecenia DDL języka SQL:

```
CREATE DATABASE "sprzedaz.sdf" DATABASEPASSWORD 'XX22Dr56'
```

2. Procedury stworzonej w dowolnym języku istniejącym dla platformy .NET, w której powołany zostanie do życia obiekt *SqlCeEngine* reprezentujący bazodanowy silnik systemu SQL Server Mobile z określonym parametrem *connectionString*. Parametr ten zawiera dane na temat lokalizacji i sposobu łączenia się z nowo tworzoną bazą danych. Samo jej utworzenie realizuje metoda *CreateDatabase*.

```
string connString;
connString = "Data Source=sprzedaz.sdf; Password=\"XX22Dr56";";
SqlCeEngine engine = new SqlCeEngine(connString);
engine.CreateDatabase();
```

Szyfrowanie mobilnych baz danych – stosowane jest wyłącznie do tych, które zostały zabezpieczone hasłem. Dostęp do zaszyfrowanej bazy danych odbywa się tylko przez podanie hasła, bez dodatkowych poleceń rozszyfrowujących. Ponieważ hasła nie da się odtworzyć, lecz jednie utworzyć nowe, jego utrata czyni bazę danych nieodtwarzalną.

Szyfrowanie bazy danych, podobnie jak ustanowienie do niej hasła, odbywa się w trakcie jej tworzenia, w obu omówionych wcześniej przypadkach. Rozszerzone zostają odpowiednio: polecenie DDL o klauzulę *ENCRYPTION ON* lub parametr *connectionstring* o frazę *Encrypt*.

- `CREATE DATABASE "sprzedaz.sdf" DATABASEPASSWORD 'XX22Dr56' ENCRYPTION ON`
- `connString = "Data Source=sprzedaz.sdf; Password=\"XX22Dr56"; Encrypt = TRUE;";`

Działania te są szczególnie istotne, gdyż bazodanowe narzędzia klienckie dedykowane urządzeniu przenośnemu, takie jak *Query Analyzer* (program isqlw30.exe), nie posiadają żadnych metod uwierzytelniania. Wynika to z faktu, że w mobilnej bazie danych nie ma tabel systemowych, w których można by przechowywać nazwy kont i ich hasła.

3. Podsumowanie

W artykule przeprowadzono analizę architektury systemu, wykorzystującego mobilne bazy danych, pod kątem mechanizmów gwarantujących jego bezpieczeństwo. Analiza ta ma stanowić wskazówki oraz wytyczne dla twórców i administratorów omawianych systemów rozproszonych. Wykazała ona, że ochrona takiego środowiska wymaga podjęcia działań na wielu jego poziomach. Obejmują one określenie praw:

- do korzystania z serwera stacjonarnego, jeśli zawiera on źródło danych dla tworzonej mobilnej bazy danych,
- do używania mobilnej bazy danych,
- przeprowadzania procesu replikacji i synchronizacji danych,
- nawiązywania komunikacji pomiędzy modułami tak rozproszonego systemu.

Szczególną uwagę skupiono na narzędziach firmy Microsoft: SQL Server 2008, SQL Server Mobile, Windows Mobile oraz Internetowych Usługach Informacyjnych (IIS). W celu przeprowadzenia testów tego środowiska zaprojektowano przykładową bazę danych dla symulującą pracę systemu mobilnej sprzedaży. Z jej pomocą testowano zabezpieczenia poszczególnych jego modułów wraz z próbami ich „złamania”. W przypadku możliwości wyboru zabezpieczeń starano się wskazać te, które najlepiej będą się w sprawdzały w badanych warunkach.

Analizowane narzędzia zostały wyposażone w różne rozwiązania, które pozwalają zabezpieczać dane funkcjonujące w systemie stosującym mobilne bazy danych. Najsłabszym ogniwem pozostaje człowiek – administrator, programista, użytkownik – w rękach którego pozostaje umiejętne wykorzystanie dostępnych możliwości.

BIBLIOGRAFIA

1. Bernard G., Ben-Othman J., Bouganim L., Canals G. i inni: Mobile Databases: A Report on Open Issues and Research Directions, ACM SIGMOD, 2003.
2. Drosatos G., Efraimidis P. S.,KarakosA A.,: Secure Mobile Database Application, http://polis.ee.duth.gr/dros/files/SecMobDB.pdf, 2009.
3. Karbowski M.: Podstawy kryptografii. Wydanie II, Helion, Gliwice 2007.
4. Neto M. C.T., Salgado A.C.: Hoarding and prefeching fro mobile databases, Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse, 2006.
5. Saha D., Chowdhury N.: A Method for Secure Query Processing in Mobile Databases, http://www.engineeringletters.com/issues_v14/issue_1/EL_14_1_20.pdf, 2009.
6. Ślusarczyk P:. Systemy mobilnych baz danych jako szczególne systemy przetwarzania rozproszonego. Bazy danych – struktury, algorytmy, metody, Praca zbiorowa pod redakcją: S. Kozielskiego, B. Małysiak, P. Kasprowskiego, D. Mrozka. ISBN 83-206-1611-5, tom 2, WKŁ, 2006.
7. Xia Y., A. Helal.: A Dynamic Data/Currency Protocol for Mobile Database Design and Reconfiguration, 31st Annual International Computer Software and Applications Conference – Cover, SAC '03: Proceedings of the 2003 ACM symposium on Applied computing, New York, 2003.
8. Local Authentication Subsystem http://msdn.microsoft.com/en-us/library/ms926467.aspx, 2009.
9. Poziom bezpieczeństwa w firmach: http://osnews.pl/poziom-bezpieczenstwa-w-firmach/, 2009.
10. SQL Server Replication. http://technet.microsoft.com/en-us/library/ms151198.aspx, 2009.
11. Threat modeling: http://msdn.microsoft.com/en-us/library/aa302419.aspx? , 2009.
12. Zbiór artykułów na temat mobilnych baz danych w download-book.net: http://download-book.net/mobile-database-pdf.html, 2009
13. Zbiór artykułów na temat mobilnych baz danych: http://www.pdfgeni.com/book/mobile-database-pdf.html, 2009

Recenzenci: Dr inż. Alina Momot
Dr inż. Tomasz Traczyk

Wpłynęło do Redakcji 16 stycznia 2010 r.

Abstract

The mobile distributed system analysis was presented in the paper. This analysis concerned mechanisms guaranteeing the data security and showed that protection of such system must be performed on many levels, including access to central and mobile databases, data replication and communication of distributed system modules. The kind of mechanisms used for this purpose depends of a given mobile distributed environment. In the research the attention was focused on the Microsoft tools: SQL Server 2008, SQL Server Mobile and IIS (Fig. 1).

They were tested with usage of the sample database which can operate in the mobile sale system (Fig. 2). This database was created in the central database and then replicated to the mobile device. During this phase privileges for data publication, its subscription and synchronization were checked. These tasks are generally performed by a set of agents, which can be executed as operating system programs or database server jobs. The rights for their execution are defined in their parameters.

In the next step the mechanisms for protection of mobile data selecting and updating were analyzed as well. Data manipulation operations can be performed from mobile database application level or from special server tools. In the first case it is the application's duty to support mechanisms for user authentication and authorization. In the second one the only techniques of protecting mobile database is its encryption and securing by password.

The research has confirmed existence of solutions for distributed mobile systems protecting and, therefore, the performed analysis can constitute a set of guidelines for developers of such products.

Adres

Katarzyna HARĘŻLAK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, katarzyna.harezlak@polsl.pl