DB2 安全：爲 DB2 資料庫鑑別開發安全外掛程式

如何設計與開發 DB2 安全外掛程式，以鑑別 DB2 資料庫中儲存的使用者與群組資訊

級別：中級

IBM DB2 專案辦公室 Gene Kligerman (gene_kligerman@ca.ibm.com)

2008 年 2 月 28 日

本文說明如何爲 Linux®、UNIX® 及 Windows® 設計並開發 DB2® 安全外掛程式，從本端或遠端 DB2 資料庫接收使用者 ID、群組或群組成員資訊。使用這個外掛程式，可以無需根據作業系統等外部鑑別機制，開發資料庫應用程式。本文隨附適用於 Windows 作業系統的安全外掛程式實作；本文爲 DB2 安全系列的文章。

前言

本文會說明特定類型的安全外掛程式設計與開發程序，這種外掛程式使用 DB2 資料庫來儲存鑑別資訊。

本文還說明在外掛程式的開發程序中，可能會遇到的技術問題及解決方法。在下載部分可以找到使用此開發程序建置的 DB2 安全外掛程式，提供二進位檔與原始檔兩種格式，方便有興趣的開發人員進行學習和延伸。

在 DB2 資料庫儲存鑑別資訊的背景資料

依預設，DB2 出貨隨附的安全外掛程式會使用下列鑑別機制，以執行鑑別：

1. 作業系統（預設）
2. 輕量型目錄存取通訊協定(LDAP)
3. Kerberos

雖然上述鑑別機制適用於大部分的 DB2 部署情境，但部分 DB2 使用者與應用程式開發人員一直希望進一步瞭解 DB2 系統在 DB2 內執行鑑別作業的能力，即在 DB2 資料庫中儲存所有 DB2 使用者與群組資訊。有關的實作，以及本文說明的設計程序，都需要符合以下需求才能進行：

- 所有鑑別資訊（使用者、密碼與群組資訊）必須儲存在資料庫中。
- 支援在本端 DB2 資料庫（DB2 實例的一部分，實例使用安全外掛程式）或遠端 DB2

資料庫（另一個實例中的資料庫，該實例可能位於不同的系統）儲存鑑別資訊。支援本端資料庫可讓您開發 DB2 資料庫以及完全自行包含的周邊應用程式（出貨時預先定義的使用者與群組清單）。支援遠端資料庫則可讓您無須部署外部服務，例如 Kerberos 或 LDAP，即可集中管理位於不同伺服器上的多個 DB2 資料庫鑑別資訊。

- 所有鑑別作業應透過 SQL 執行，讓您可以撰寫外部 DB2 應用程式，在本端或遠端管理這項資訊。
- 鑑別資訊應使用安全的方法儲存，標準應與 DB2 安全哲學一致。
- 實作不應使用內部 API，並應提供程式碼及設計資訊，讓面對類似問題的使用者可以輕易取得資訊，並加以延伸。

外掛程式安裝與配置概觀

若要充分瞭解本文所述的 DB2 安全外掛程式，最好的方法是檢閱外掛程式在安裝與配置方面的重要特色。

第一步是在外掛程式配置檔中，指定鑑別資料庫的名稱，以及使用者的連線資訊。外掛程式會使用資訊來連結資料庫，以進行所有後續的鑑別要求，鑑別資料庫可以位於本端或遠端。如果資料庫位於遠端，建議使用資料加密來配置遠端資料庫，以提高安全性，也即是開啓 **data_encrypt** 配置參數。

第二步是將外掛程式配置檔連同外掛程式二進位檔，移至 DB2 伺服器外掛程式安裝目錄，您可以配置 DB2 引擎來尋找這些檔案。

第三步是在鑑別資料庫中建立綱目，以及網目中的相關資料庫物件。這些物件是內含鑑別資訊的表格，以及封裝了伺服器邏輯的儲存程序，有關儲存程序可在資料庫表格作業。

第四步是在上述建立的表格中，移入第一個使用者的鑑別資訊，即使用者 ID，以及對應於外掛程式配置檔指定的使用者 ID 密碼資訊。

完成上述步驟之後，DB2 伺服器管理者只需變更管理程式配置參數，指示以 DB2 鑑別安全外掛程式，取代預設的作業系統外掛程式。

重新啓動 DB2 資料庫實例之後，就會使用新的外掛程式。當然，資料庫中只擁有單一鑑別使用者 ID 並不足夠。在此階段，管理者可以在鑑別資料庫中新增使用者帳戶、新增群組，並指派使用者爲群組成員。

管理者也可以讓其他使用者擔任鑑別資訊的管理者，只需在資料庫表格與儲存程序中，使用 SQL 授與適當的專用權即可。

在外掛程式下載中，另有提供完整的說明文件，講解安裝、配置及外掛程式的使用。

DB2 鑑別外掛程式程序模型

全新 DB2 安全外掛程式的起點是名爲 combined.c 的 DB2 安全外掛程式範例，範例在出貨時隨附於已安裝的 DB2 sqllib/samples/security 目錄中。combined.c 是很簡易的外掛程式，可以從預先定義的純文字檔讀取鑑別資訊（使用者 ID、明碼密碼及群組成員）。因此，先前瑣碎的作業可以彙整如下：以 SQL 介面取代檔案存取的呼叫，因爲它會存取 DB2 資料庫的存取資訊，而非純文字檔。

實現外掛程式開發其實並不瑣碎，影響外掛程式設計的主要因素，是外掛程式做爲 DB2 引擎程序 db2syscs.exe 之部分而運作。由於不可能在 DB2 引擎程序的環境定義中載入 DB2 應用程式，因此，所有 SQL 互動必須在 db2syscs.exe 以外的程序中執行。

故此，新的 db2auth 外掛程式將使用另外的程序實作，db2auth.c 檔包含了已鏈結 DB2 引擎程序的程式碼，db2auth.c 在外掛程式起始設定中，將會建立雙向的具名管線，接著建立並啓動另一個常駐程式處理程序，該程序將執行 SQL 與 DB2 資料庫之間的所有互動，而該程序的程式碼則在 db2authDaemon.c 檔中實作。這個常駐程式處理程序透過具名管線，從 DB2 引擎接收鑑別要求，使用 DB2 呼叫層次介面 (CLI) API 查詢 DB2 資料庫，接著透過相同的管線機制，將結果傳回 db2auth 外掛程式。

對於單一要求而言，這種方法可以順利進行，但 DB2 for Windows 是多執行緒引擎（DB2 for Unix 及 DB2 for Linux 9.5+ 也是如此），因此，對於提出鑑別要求的多重並行引擎執行緒，外掛程式必須正確處理引擎執行緒的執行。爲了解決這個問題，可以將外掛程式與常駐程式之間的所有互動序列化，讓外掛程式每次只可以處理一個鑑別要求。在 Windows 平台上，使用 CreateMutex API 即可完成這個設定。

透過兩個具名管線（一個用於所有的使用者鑑別要求，另一個用於所有的群組鑑別要求），對鑑別要求進行序列化的效能又如何？一項非科學的實作結果效能分析顯示，外掛程式每分鐘能夠處理超過 1,000 個連線要求，甚至可以有更好的效能。這個測試是在 Windows XP 雙核心筆記型電腦上執行，使用 DB2 9.1 版程式碼庫，並模擬數個 DB2 指令行處理器工作串流，其中只包含 **connect** 陳述式。

db2authDaemon 程序是在外掛程式起始設定中啓動（例如在 db2start 處理期間），常駐程式處理程序可能在一般外掛程式終止（以具名管線的訊息終止），或者在得知母程序已終止時即會終止。

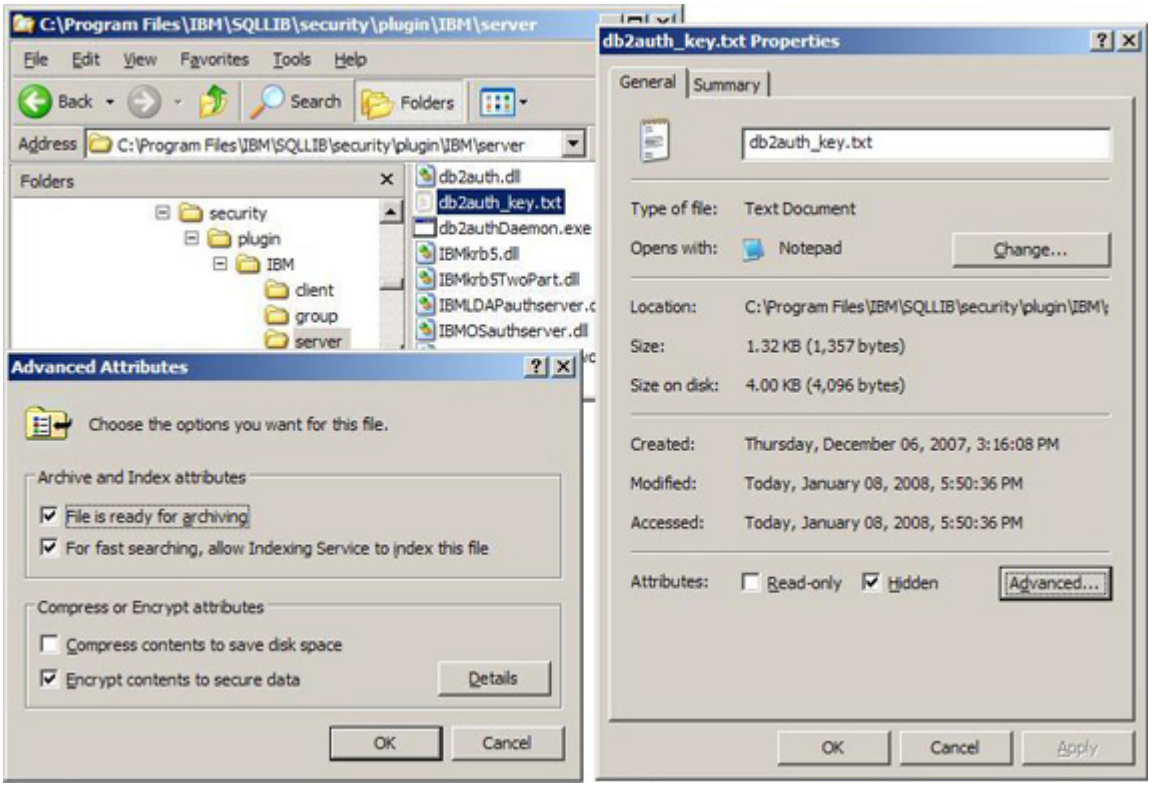
閱讀過程序模型的介紹之後，您現在可以進一步鑽研外掛程式的功能了。

外掛程式運作

第一個鑑別要求傳入時，DB2 外掛程式會與內含鑑別資訊的 DB2 資料庫連線。爲了避免發生遞迴，外掛程式必須擁有充分的連線資訊，例如 DB2 資料庫名稱（資料庫別名）、資料庫授權使用者的使用者 ID 及（選用）連線密碼，

DB2 實例管理者必須在名爲 db2auth_key.txt 的純文字檔中提供這類資訊。如果 DB2 鑑別資料庫及外掛程式位於相同的 DB2 實例，請勿指定密碼，以進行隱含連線；然而，假如資料庫位於不同實例（很可能是不同的伺服器），則務必提供連線密碼。在檔案中以明碼儲存密碼，可能造成安全漏洞，DB2 管理者應盡量避免。在 Windows 平台上，有多個簡易方法可以保護這個檔案的內容，例如變更檔案屬性，讓 DB2 實例擁有者程序存取檔案的同時，也隱藏與加密檔案，如圖 1 所示。

圖 1. 保護連線資訊檔的安全



在這個純文字檔案中，其他需要指定的資訊只剩已配置的 DB2 System Administration 群組名稱（透過 DB2 Database Manager SYSADM_GROUP 配置參數），此舉可確保 DB2 實例擁有者 ID 享有 SYSADM 專用權。由於只能在檔案中指定一個群組名稱，所以啓用外掛程式之後，針對 DB2 實例擁有者 ID 所進行的群組成員資格檢查只會傳回該群組名稱。按照目前的實作，外掛程式的部署將 SYSADM_GROUP 配置參數設定成外掛程式部署階段中所指定的值。相較於 DB2 出貨時預設在 Windows 平台的 O/S 型外掛程式，這做法可以提供更大的

好處。SYSADM 群組無需在作業系統中進行定義，同時也不需要擁有 Windows 管理者專用權。

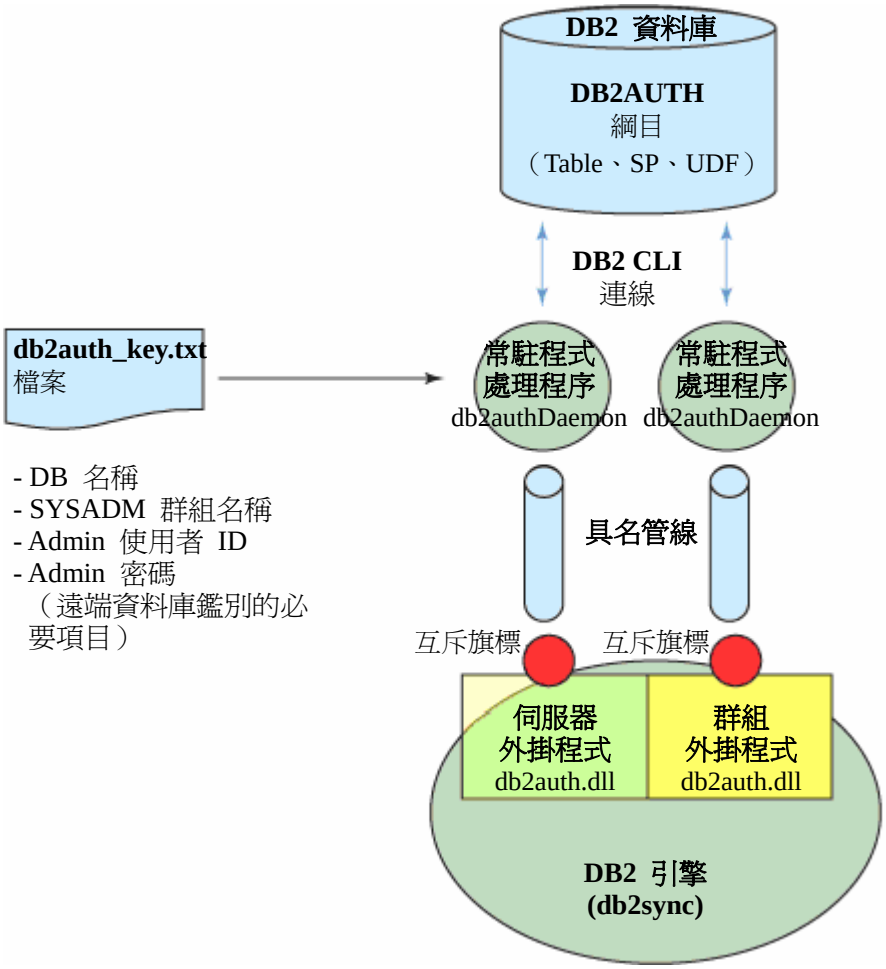
外掛的常駐程式與 DB2 鑑別資料庫之間有持久的連線，即在常駐程式作業期間都會維持連線。可想而知，若連線意外終止，可能導致鑑別要求無法繼續，爲了盡量降低這種可能性的影響，常駐程式會在後續處理，嘗試重新連線至鑑別資料庫，重新建立連線之前，一般使用者會收到錯誤碼，指出鑑別要求失敗。若在外掛程式處理期間發生這些事件，診斷資訊將寫入至 %db2instprof%\DB2instanceName\db2diag.log 檔，而部分外掛程式錯誤訊息將顯示於 Windows 事件日誌。

若是本端的鑑別資料庫，發出 **db2stop** 指令以關閉實例的動作失敗時，會出現訊息，顯示資料庫仍然作用中，因爲外掛程式的資料庫連線尚未終止。因此，使用本端鑑別資料庫來部署此外掛程式時，必須使用 **db2stop force** 指令來關閉實例。管理者也可以使用 **db2 list applications** 指令，來檢查作用中的應用程式，並使用 **db2 force applications** 指令，終止該群組與安全外掛程式所建立的資料庫連線。這些資料庫連線可以名稱進行識別：db2AuthDaemon.exe，這是 DB2 鑑別外掛程式與預設的 O/S 外掛程式之間，唯一明顯的作業差異。

db2auth 外掛程式實作遵循預設的 DB2 哲學，假設在授信環境的作業系統上執行。若使用者並未指定密碼，並嘗試連線至本端資料庫（即在相同伺服器與部分相同的 DB2 實例上）時，在假定使用者已由作業系統進行鑑別的前提下，將容許該連線。此機制稱爲「DB2 隱含本端登入」，當然也可以修改外掛程式碼，在本端使用者連線時要求提供密碼，但這樣將停用部分功能的作業，例如「DB2 性能監控」。

圖 2 說明 DB2 安全外掛程式的架構：

圖 2. 外掛程式架構



資料庫結構

此部分將介紹 DB2 資料庫物件（例如：綱目、表格、索引與預儲程序），以及這些物件的應用程式介面，

爲安全外掛程式而建立的所有資料庫物件都會封裝在 DB2AUTH 綱目中。爲外掛程式使用專用綱目，可以將安全資訊與其他的資料分開，以在實務上使用單一資料庫同時包含鑑別資訊與其他資訊。

您需要建立三個表格以維護鑑別資訊：

- *USERS* 表格包含使用帳戶資訊，其中包括 DB2 authid（鑑別 ID）、密碼、密碼到期日、帳戶狀態，以及登入失敗次數。

在此實作中，資料庫絕對不會以明碼儲存密碼，而且外掛程式也不會以明碼傳送密碼。外掛程式將使用 MD5 雜湊演算法，雜湊明碼密碼，並且以此雜湊形式儲存於資料庫。驗證密碼時，會比較使用者所提供密碼的雜湊簽章及資料庫中儲存的簽章。

外掛程式會使用密碼到期日，判斷密碼是否現行可用。本實作的密碼到期日，是密碼起始建立或變更以後的 90 天。90 天之後，密碼驗證將會失敗，若要讓使用者通過帳戶鑑別，唯一的方法是由管理者重設使用者的密碼（會同時重設 90 天時限）。

管理者可以使用帳戶狀態旗標，暫時啓用或停用該帳戶，預設的帳戶狀態是 'N'（正常），若提供有效密碼，即可成功通過鑑別。不過，管理者也可以將帳戶狀態旗標變更爲 'L'（鎖定），如此一來，即使提供正確密碼，鑑別要求仍會遭到拒絕。

失敗登入計數器將會計算錯誤密碼導致的鑑別要求失敗次數。如果使用者提供錯誤密碼的次數超過四次，將會鎖定帳戶直到管理者重設爲止；不過，如果使用者在第二次、第三次或第四次提供正確密碼，就會成功通過鑑別要求，而計數器也將重設爲 0。

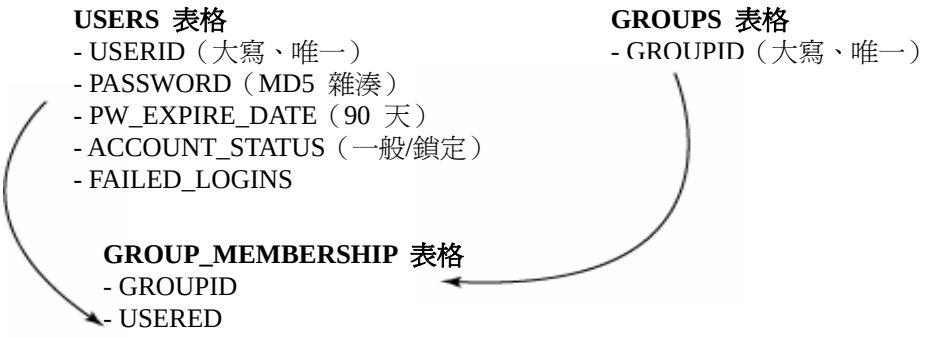
- *GROUPS* 表格包含管理者已建立的 DB2 群組。

- *GROUP_MEMBERSHIP* 表格包含使用者 ID 隸屬群組的資訊，每一列都是值組，內含組群名稱及使用者 ID。這個表格使用參照完整性限制，確保資料的有效性，因此，如果刪除了 USERS 表格中的某個使用者 ID，GROUP_MEMBERSHIP 表格也會刪除該項目。同樣地，如果刪除了 GROUPS 表格中的某個群組，GROUP_MEMBERSHIP 表格也會刪除該群組。

  就內部而言，DB2 處理使用者 ID 與群組名稱時，也會考量其中的大小寫，這表示外掛程式所收到的使用者 ID 與群組名稱可能是小寫（例如，使用者在應用程式中輸入密碼）或大寫（例如，DB2 參照內部專用權檢查機制），但在資料庫中，所有使用者 ID 與群組名稱都會以大寫方式儲存。

圖 3 是上述資料庫物件的圖解：

圖 3. 鑑別資料庫的設計



SQL 應用程式介面

外掛程式架構已經設計好了，而資料庫設計也完成了，本部分將介紹應用程式介面。

外掛的常駐程式是與鑑別資料庫進行通訊的主要應用程式，由於使用容易，所以會使用 CLI。請注意，常駐程式與資料庫之間的互動，絕對不能觸發資料庫進行其他鑑別。資料庫引擎會把鑑別要求遞送至外掛程式，因此在處理前一個鑑別要求時，可能導致死鎖，結果，可能造成常駐程式無法呼叫 SQL 儲存程序之類的問題。

外掛的常駐程式與鑑別資料庫之間的互動很簡單，以下是部分常駐程式發出的 SQL 陳述式。

清單 1. 重設使用者的密碼

```
UPDATE db2auth.users SET failed_logins = 0, password = ?,
                    pw_expire_date = (current date + 90 days)
                WHERE USERID = UPPER(?)
```

清單 2. 取得使用者所屬的群組清單

```
SELECT groupid FROM db2auth.group_membership WHERE userid = ?
```

清單 3. 在太多無效鑑別嘗試之後鎖定帳戶

```
UPDATE db2auth.users SET account_status = 'L', failed_logins = failed_logins + 1
                WHERE userid = UPPER(?)
```

以上程式碼都會處理安全外掛程式，不過，完整的解決方案需要提供介面，以供擁有資料庫資訊的安全管理者使用。您可以使用 C/C++、Java、Perl、PHP 或其他介面來開發管理工具，但使用 SQL 儲存程式是封裝管理應用程式商業邏輯的最佳方式，可將開發人員需要撰寫的程式碼數量減至最少。

以下是已經完成撰寫的 SQL 儲存程序，儲存程序使用 DB2 CLP 可直接執行的格式呈現：

- 重設使用者的帳戶：
  db2 call db2auth.change_password_admin('kohlmann', '<舊的密碼雜湊>', '<新的密碼雜

湊>', ?, ?, ?)

- 新增使用者：
  db2 call db2auth.add_user('gene', '<密碼的 MD5 雜湊簽章>', ?, ?, ?)
- 新增使用者群組：
  db2 call db2auth.add_group ('db2users', ?, ?, ?)
- 刪除使用者
  db2 call db2auth.del_user('gene', ?, ?, ?))
- 刪除群組
  db2 call db2auth.del_group('db2users', ?, ?, ?)
- 讓使用者成爲群組成員：
  db2 call db2auth.add_groupmember('db2admns', 'gene', ?, ?, ?)
- 將使用者從群組成員中移除：
  db2 call db2auth.del_groupmember ('db2users', 'gene', ?, ?, ?)

清單 4 是一項儲存程序的清單：

清單 4. change_password_admin SQL 儲存程序的程式碼

```
CREATE PROCEDURE CHANGE_PASSWORD_ADMIN (IN userid_in VARCHAR(255),
                                IN newpassword_in VARCHAR(16) FOR BIT DATA,
                                OUT sqlstate_out CHAR(5),
                                OUT sqlreason_out INTEGER,
                                OUT message_out VARCHAR(70))
         DYNAMIC RESULT SETS 1
-------------------------------------------------------------------------
-- SQL Stored Procedure
         -- userid_in
         -- newpassword_in (hashed using MD5 algorithm)
         -- sqlstate_out:  Security processing error will result in '08001'
         -- sqlreason_out:  Error reason code.  See reason codes associated with
                        SQLSTATE 08001 in DB2 doc
         -- message_out: Error message text
-------------------------------------------------------------------------
P1: BEGIN
         -- Declare variables
         DECLARE SQLSTATE CHAR(5) DEFAULT '00000';
         DECLARE SQLCODE INT DEFAULT 0;
         DECLARE valid_userid SMALLINT DEFAULT 1;
```

```
        -- Declare handler
        DECLARE CONTINUE HANDLER FOR NOT FOUND
                 SET valid_userid = 0;

        SET sqlstate_out = '00000';
        SET sqlreason_out = 0;
        SET message_out = '';

        IF userid_in = '' THEN
                 SET sqlstate_out = '08001';
                 SET sqlreason_out = 6;
                 SET message_out = 'Security processing failed: Invalid userid.';

        ELSE -- try to change the password
                 UPDATE DB2AUTH.USERS SET failed_logins = 0, account_status = 'N',
                                          PASSWORD = newpassword_in
                   WHERE USERID = UPPER(userid_in);
        END IF;

        IF valid_userid = 0 THEN -- userid not found in user table, return error
                 SET sqlstate_out = '08001';
                 SET sqlreason_out = 24;
                 SET message_out = 'Security processing failed: Invalid userid.';
        END IF;

END P1
```

使用者介面

建立以上的基礎架構之後，則欠缺一項元素，即是讓 DB2 使用者及系統互動，以進行鑑別與管理的機制。

管理使用者以外的使用者只需要執行兩種基本作業的能力：鑑別使用者 ID/密碼組合與變更他們的密碼，而 DB2 產品中的 CONNECT API 已有提供這些功能。例如，以下的 DB2 CLP 指令爲使用者 GENE 變更密碼的同時，也嘗試與資料庫 DB2AUTH 建立連線：

清單 5. 使用 DB2 CLP 變更密碼

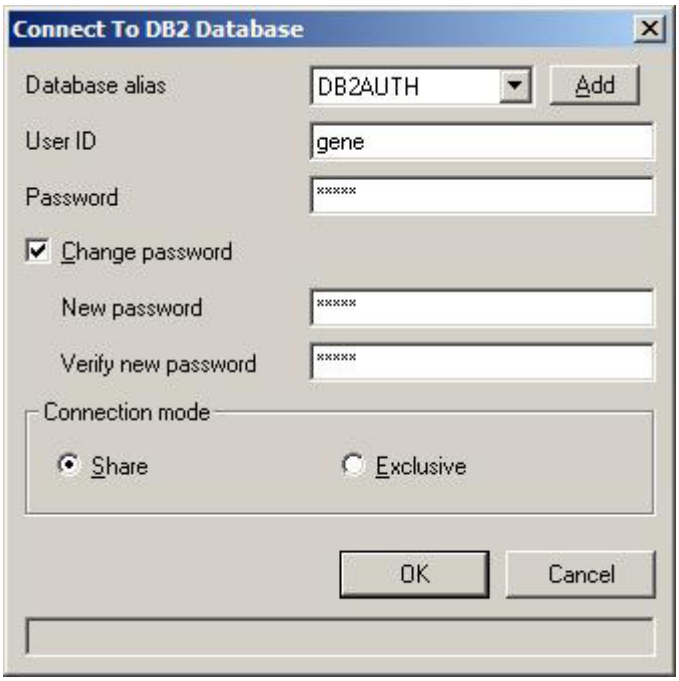
```
C:\>db2 connect to DB2AUTH user GENE change password
Enter current password for GENE:
Enter new password for GENE:
Confirm new password for GENE:

   Database Connection Information

 Database server       = DB2/NT 9.5.0
 SQL authorization ID   = GENE
 Local database alias   = DB2AUTH
```

當然，此功能不限於 CLP ，在所有 API 中，凡是 DB2 支援的使用者撰寫應用程式，都可以執行這項功能。例如，使用 CLI API 在 Windows 上執行的 C 或 C++，會呼叫 SQLDriverConnect API，如下所示：

```
SQLDriverConnect (hdbc, (SQLHWND)NULL, "DSN=;UID=;PWD=;", SQL_NTS,
         NULL, 0, NULL, SQL_DRIVER_PROMPT)
```

執行包含此程式碼的應用程式將啓動以下的 GUI 介面：

圖 4. 資料庫連線介面



鑑別資料的管理者需要應用程式介面，以管理使用者帳戶與群組資訊，這個應用程式介面不需要很複雜或很精細，除了開發管理介面的展現層（例如：使用 Java、PHP 或 Windows 原生 API，開發 GUI 介面）之外，介面的 SQL 層可以極爲簡單，介面 SQL 層主要的作業就是呼叫上述的 SQL 儲存程序。

未來可能提供的加強功能

有興趣的開發者，可以將現行的外掛程式實作多方面延伸。

起始實作以二進位形式出貨，僅適用於 32 位元 Windows 平台，相較於重新編譯外掛 C 程式碼以產生 64 位元程式碼，製作可以在 64 位元 Windows 平台上運作的外掛程式，後者應該更爲簡單。在 Linux 及 Unix 平台上植入外掛程式碼會比較困難，因爲其中的跨處理程序通訊 (IPC) 與 MD5 雜湊簽章會使用 Windows 專用 API。

另一個有待加強的領域是針對使用者密碼，實作其他規則，例如，可在外掛程式中新增其他邏輯，以加強密碼長度下限與密碼的複雜性。也可以很簡單地在解決方案新增其他表格，爲特定使用者 ID 儲存之前 N 次的密碼雜湊簽章。使用者要求變更密碼時，可以檢查表格以查看新密碼是否與最近 N 次使用的密碼相同。

雖然使用雜湊法可以確保外掛程式，不會使用明碼在網路上傳送密碼，或者在資料庫中儲存密碼，但您還是可以實作其他技術，以降低惡意攻擊者猜到密碼值的可能性，其中一種方式就是使用變化數值 (salt)。變化數值可在混合密碼之前先加入其他資料，讓攻擊者無法使用預先運算的字典。

在使用性方面，極度需要開發 GUI 介面，以管理鑑別資訊資料庫，現行的外掛程式實作在出貨時，已隨附以字元爲主的互動介面，但 GUI 並不具備使用者友善特性。其中一種可能的方法，是開發以 Java 撰寫的獨立式管理介面；另一種方法則是開發新的建置區塊，然後插入以 PHP 撰寫的 DB2 Monitoring Console。DB2 Monitoring Console 是由建置區塊組成的開放程式碼專案，可快速開發您自己的主控台或 PHP Web 型應用程式。

總結

本文介紹了使用 SQL 介面的 DB2 安全外掛程式的設計，以存取 DB2 資料庫中儲存的鑑別資訊。若要下載本文所述的程式碼（提供二進位檔與原始檔兩種格式），請參考「下載」部分。

鳴謝

本專案全賴以下人員的協助，方可完成：

- Peter Kohlmann，他從專案的概念到開發過程中，都提供了鼓勵及支援
- Scott Logan，他針對如何架構 DB2 安全外掛程式，分享了見解
- Garfield Lewis，他提供了 Windows 應用程式開發的要訣
- Kevin See、Henry Chan、Walid Rjaibi 與 DB2 開發部門的其他人員，他們針對解決方案成品提供了寶貴的意見

下載

說明	名稱	大小	下載方法
適用於 Windows 的安全外掛程式實作	db2auth.zip	60KB	HTTP

下載方法的相關資訊

資源

Learn

- developerWorks resource page for DB2 for Linux, UNIX, and Windows: Read articles and tutorials and connect to other resources to expand your DB2 skills.
- Learn about DB2 Express-C, the no-charge version of DB2 Express Edition for the community.
- "Understanding DB2 9 Security" (ISBN: 0131345907): Get a wealth of security information that isn't available anywhere else, direct from a DB2 security deployment expert and the IBM DB2 development team.

- DB2 Monitoring Console: Find the open source project that can be used to monitor DB2 and is easily extensible.
- developerWorks Information Management zone: Learn more about DB2. Find technical documentation, how-to articles, education, downloads, product information, and more.
- Stay current with developerWorks technical events and webcasts.

Get products and technologies

- Build your next development project with IBM trial software, available for download directly from developerWorks.

Discuss

- Participate in the discussion forum.
- Participate in developerWorks blogs and get involved in the developerWorks community.

關於作者

Gene Kligerman 自 1987 年開始在 IBM 任職，曾經擔任產品開發、技術行銷與版本管理的職務，目前在 IBM 加拿大多倫多實驗室的 DB2 專案辦公室工作。