УДК 004.056.5:004.416

Г.А. Гурвиц,
канд. техн. наук, доцент
Дальневосточного государственного университета путей сообщения
(г. Хабаровск)
Р.А. Ешенко,
канд. техн. наук, доцент
Дальневосточного государственного университета путей сообщения
(г. Хабаровск)

ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ КОРПОРАТИВНЫХ ПРИЛОЖЕНИЙ

Предложен новый способ создания формы, обеспечивающий контроль доступа клиентской части корпоративных приложений.

Ключевые слова: *информационная безопасность, контроль доступа, корпоративное приложение.*

A new way to create a form that provides access control to the client-side of corporate applications is proposed.

Keywords: *information security, access control, corporate application.*

Чем большую ценность представляет собой информация, находящаяся в распоряжении предприятия, тем острее стоят вопросы его доверия к своим сотрудникам и тем совершеннее должны быть средства отслеживания доступа к корпоративным информационным ресурсам, бесконтрольное использование которых может причинить серьёзный вред. Любое предприятие не может сегодня успешно функционировать без создания надёжной системы защиты информации, включающей не только организационно-нормативные меры, но и технические программно-аппаратные средства контроля [1]. Все решения по защите от несанкционированного доступа и установлению личной ответственности работников принимаются руководителем предприятия, так как именно он по действующему ныне законодательству несёт в первую очередь ответственность за утрату конфиденциальной информации. При защите информационной системы от несанкционированного доступа могут быть использованы два принципа управления доступом к защищаемым ресурсам – дискреционный и мандатный [2].

При дискреционном принципе каждому зарегистрированному пользователю устанавливаются права доступа к объектам системы, которые прописываются в правилах разграничения доступа (ПРД). Эти правила должен утвердить руководитель предприятия. Данный вариант управления доступом позволяет для любого пользователя системы создать изолированную

программную среду, то есть ограничить его возможности по запуску программ, указав в качестве разрешённых к запуску только те программы, которые действительно необходимы для выполнения пользователем своих служебных обязанностей. Таким образом, программы, не входящие в этот список, пользователь запустить не сможет.

Мандатный принцип управления доступом к ресурсам основан на сопоставлении уровня конфиденциальности каждого ресурса и полномочий конкретного зарегистрированного пользователя по доступу к ресурсам информационной системы с заданным уровнем конфиденциальности. Для организации мандатного управления доступом для каждого пользователя системы устанавливается некоторый уровень допуска к конфиденциальной информации (уровень отдела, уровень департамента, уровень дирекции и т. д.), а каждому ресурсу присваивается так называемая метка конфиденциальности, или код доступа. При этом разграничение доступа к информации осуществляется путём сравнения уровня допуска пользователя и метки конфиденциальности ресурса и принятии решения о предоставлении или непредоставлении доступа.

Рассмотрим пример формы, представляющий дискреционный принцип управления доступом, который без каких-либо переделок сможет реализовать и мандатный принцип управления доступом. Наряду с контролем доступа система защиты программного комплекса содержит подсистему учёта сделанных пользователем изменений. Как только пользователь получит право доступа к работе с программным комплексом, ему автоматически предоставляются различные привилегии. Они могут включать разрешение на доступ к определённым таблицам, формам, запросам, отчётам и расчётам, которые выполняет комплекс. Привилегии предоставляются работникам для того, чтобы они могли решать задачи, входящие в круг их должностных обязанностей. К нарушению защиты может привести предоставление излишних полномочий, поэтому каждому работнику администратор должен назначить только те привилегии, без которых его работа невозможна. Рассматриваемая форма делает доступными каждому пользователю только те пункты меню и кнопки, с которыми ему положено работать в соответствии ПРД, существующими на предприятии. Она запускается на выполнение до активации меню программного комплекса и либо вообще запрещает регистрацию работника, если его данных (фамилия и пароль) нет в базе, либо ограничивает доступ в соответствии с ПРД.

На рисунке 1 показан вид формы **Log-in** в конструкторе. В окне свойств несколько десятков строчек. Все содержат значения по умолчанию. В нашем случае изменены всего несколько из них. Список изменённых свойств приведён в таблице 1. К несомненным достоинствам MS Access при работе с

окном свойств следует отнести русифицированный интерфейс разработчика, а к недостаткам – отсутствие выделения свойств объектов, которые были изменены разработчиком. В MS Access постоянно наступают различные события, которые можно считать полезными лишь в том случае, если знаешь, как на них реагировать и в каком порядке эти события выполняются. Более того, каждый тип элементов управления Access имеет свой набор событий. Обработать событие и отреагировать на него позволяет вкладка окна свойств формы «События». При загрузке формы выполняется следующая цепочка событий:

– открытие (open);

– загрузка (load);

– изменение размера (resize);

– включение (activate);

– получение фокуса (gotfocus).

Для двух из них **Открытие** (Form_Open) и **Загрузка** (Form_Load) написан код «Процедура обработки событий», который запускается на выполнение при наступлении события. Это листинги 1 и 2.

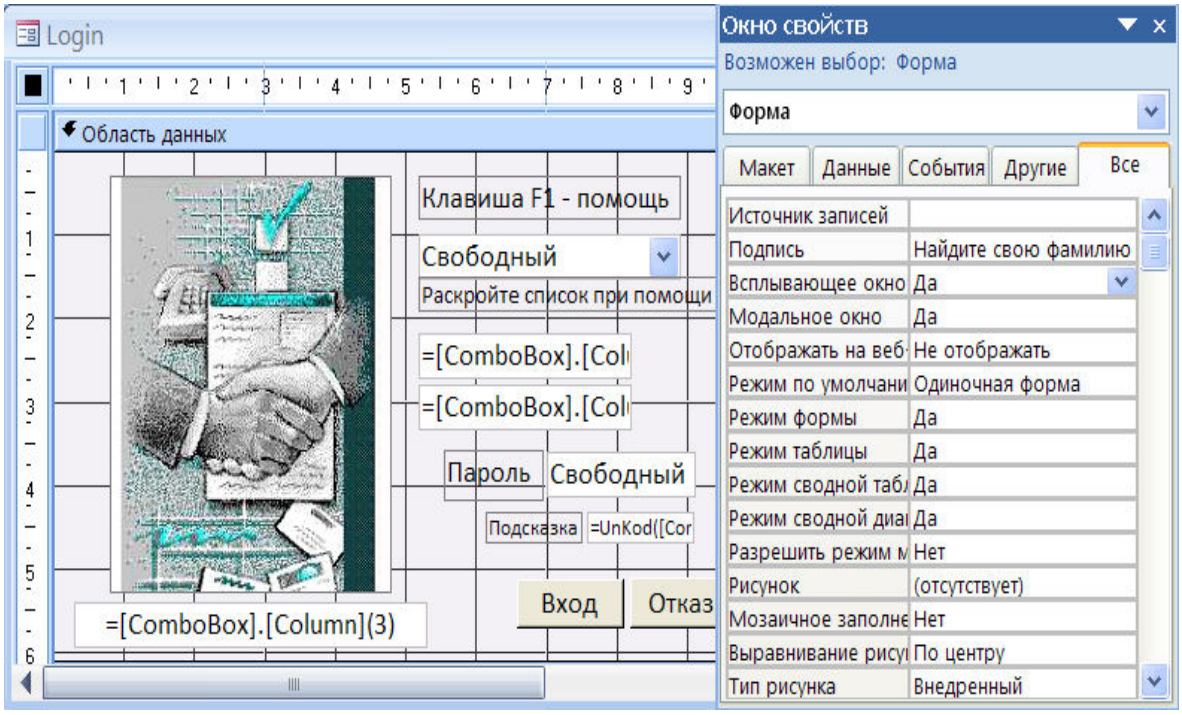

Рисунок – Форма контроля доступа к программному комплексу в режиме конструктора

Таблица 1 – Список изменённых свойств формы **Login**

Номер	Свойство	Значение
1	Подпись	Найдите свою фамилию в списке
2	Всплывающее окно	Да
3	Модальное окно	Да

4	Отображать на веб-узле Share Point	Не отображать
5	Разрешить режим макета	Нет
6	Область выделения	Нет
7	Кнопки перехода	Нет
8	Полосы прокрутки	Отсутствуют
9	Кнопки размеров окна	Отсутствуют
10	Кнопка закрытия	Нет
11	Контекстные меню	Нет

Листинг 1. Текст события «Открытие» формы **Login**

```
Private Sub Form_Open(Cancel As Integer)
' Инициализация глобальных переменных,
' предназначенных для управления доступом к приложению
' Запуск процедуры
 Adjustment ' Процедура из модуля ModuleMain
End Sub
```

Листинг 2. Текст события «Загрузка» формы **Login**

```
Private Sub Form_Load()
 ' Назначение файла справки
 Me.HelpFile = CurrentPath() & "\RealEstate.chm"
End Sub
```

RealEstate.chm – имя файла контекстно-зависимой справки к приложению. Функция **CurrentPath()** возвращает полный путь к этому файлу. Текст процедуры **Adjustment** находится в модуле **ModuleMain** и приведён в листинге 3.

Листинг 3. Текст процедуры «Настройки»

```
Public Sub Adjustment()
' Начальные значения глобальных переменных
 FAMILY = "Разработчик комплекса"
 SuperVisor = False ' Признак идентификации
 ChangePicture = False ' Смена пароля
 ChangePassword = False ' Смена пароля
 Staff = False ' Администрирование
End Sub
```

Форма **Login** работает с таблицей **tblUser**, которая включена в базу данных **Real Estate.** Состав полей таблицы **tblUser** приведен в таблице 2. Первичный ключ таблицы создан при помощи связки полей **LastName + FirstName + SecondName**. В основе его создания лежит предпосылка, что на предприятии не могут работать люди с полностью совпадающими фамилией, именем и отчеством. Если это вас не устраивает, воспользуйтесь дополнительным полем – счетчиком.

Таблица 2 – Состав таблицы пользователей

№	Поле	Тип	Размер	Описание
1	LastName	Текстовый	15	Фамилия пользователя
2	FirstName	Текстовый	12	Имя
3	SecondName	Текстовый	10	Отчество

4	Post	Текстовый	25	Занимаемая должность
5	PassWord	Текстовый	10	Зашифрованное значение пароля
6	File	Текстовый	8	Имя файла с фотографией
7	Access01	Логический	1	Настройка картинки в главном окне
8	Access02	Логический	1	Возможность смены своего пароля
9	Access03	Логический	1	Показ удалённых записей
20	Access14	Логический	1	Установка прав доступа всем работникам
21	Inspector	Текстовый	15	Фамилия предоставившего доступ
22	Date_up	Дата/Время	8	Дата последней корректировки
23	Time_up	Текстовый	10	Время последней корректировки
24	Range	Числовой	1	Типовой код доступа

Значения полей **Access01** – **Access14** присваиваются глобальным переменным в результате работы формы **Login**, и на основании того, **False** это или **True**, обеспечивается доступ к пунктам меню, формам и расчётам программного комплекса. Доступ к пункту меню обеспечивается использованием одной строчки. В листинге 4 таких строчек две (по числу пунктов меню), и они выделены жирным цветом:

Листинг 4. Доступ к пунктам меню

```
With .Add(Type:=msoControlPopup)
.Caption = "Справочники"
 With .Controls
 With .Add(Type:=msoControlButton)
.Caption = "Улицы города"
 Enabled = StreetTown
 OnAction = "Street"
 End With
 With .Add(Type:=msoControlButton)
 Caption = "Районы города"
 Enabled = DistrictTown
.OnAction = "District"
 End With
 End With
 End With
```

Управление доступом к справочнику улиц обеспечивает глобальная переменная **StreetTown**, а к справочнику районов – **DistrictTown.** Доступ к объектам формы обеспечивает это же свойство – **Enabled**, только установленное для объекта формы. В листинге 5 показано, как «связать» кнопку занесения новой записи **CommandAdd** с глобальной переменной **AddBuilding**. Установите это свойство при загрузке формы:

Листинг 5. Доступ к объектам формы

```
Private Sub Form_Load()
CommandAdd.Enabled = AddBuilding
End Sub
```

Для текстового поля **txtFirstName** (имя работника) в качестве значения свойства «Данные» необходимо указать:

=[ComboBox].[Column](1).

Колонки поля со списком MS Access нумерует с нуля. Нулевая колонка – фамилия, первая – имя, вторая – отчество и т.д. Теперь имя работника – связанный элемент. Как только в поле со списком **ComboBox** будет выбран очередной работник, в текстовом поле **txtFirstName** появится его имя. Для отображения фотографии работника в объекте формы (свободная рамка объекта) с именем **PictureUser** в качестве свойства «Данные» требуется поставить:

=CurrentPath() & "\PHOTO\" & [ComboBox].[Column](5) & ".jpg".

Составные части этого выражения:

– функция **CurrentPath()** возвращает полный путь к файлу базы данных и описана в предыдущей главе;

– PHOTO – папка, в которой хранятся фотографии работников;

– в пятой колонке поля со списком содержится значение шестого поля таблицы **tblUser**. Это поле **File** (имя файла с фотографией);

– JPG – формат файла изображения.

Текстовый элемент **txtHint** (подсказка) связан с четвёртой колонкой поля со списком:

=UnKod([ComboBox].[Column](4)).

В реальном приложении он, без сомнения, должен быть опущен. Не следует удалять этот объект из формы. Просто погасите его. Для этого в качестве значения свойства «Вывод на экран» поставьте «Нет». Свойство расположено на вкладке «Макет». Пароль пользователя хранится в зашифрованном виде в поле **PassWord** таблицы **tblUser**. Для его зашифровки используется функция **CrKod**, а для расшифровки **UnKod**. Тексты функций приведены в листингах 6 и 7.

Листинг 6. Зашифровка пароля.

```
Public Function CrKod(cPassword) As String
 ' Возвращает зашифрованный пароль
 ' cPassword - незашифрованный пароль (входной параметр)
 Dim cLetter As String ' Один символ пароля
 Dim cEncryptedPassword As String ' Зашифрованный пароль
 Dim i As Integer ' Параметр цикла
 If Not IsNull(cPassword) Then
 ' Пустой пароль не зашифровывается
 ' Убираем концевые пробелы в пароле
 cPassword = Trim(cPassword)
 ' Начальное значение зашифрованного пароля
 cEncryptedPassword = ""
 For i = 1 To Len(cPassword)
 ' Отделяем очередной символ
 cLetter = Mid(cPassword, i, 1)
 cEncryptedPassword = cEncrypt-
edPassword + _
 Chr(Asc(cLetter) + i)
 Next i
 End If
 ' Возвращаемое значение зашифрованного пароля
 CrKod = cEncryptedPassword
End Function
```

Листинг 7. Расшифровка пароля

```
Public Function Un-
Kod(cEncryptedPassword) As String
 ' Возвращает расшифрованный пароль
 'cEncryptedPassword - зашифрованный пароль
 ' (входной параметр)
 Dim cLetter As String ' Один символ пароля
 Dim cPassword As String ' Расшифрованный пароль
 Dim i As Integer ' Параметр цикла
 If Not IsNull(cEncryptedPassword) Then
 ' Пустой пароль не расшифровывается
 'Убираем концевые пробелы в зашифрованном пароле
 cEncryptedPassword = Trim(cEncryptedPassword)
 ' Начальное значение расшифрованного пароля
 cPassword = ""
 For i = 1 To Len(cEncryptedPassword)
 ' Отделяем очередной символ
 cLetter = Mid(cEncryptedPassword, i, 1)
 cPassword = cPassword + Chr(Asc(cLetter) + i)
```

```
    Next i
    End If
    ' Возвращаемое значение расшифрованного пароля
    UnKod = cPassword
End Function
```

В тексте используются стандартные функции VBA:

– Mid() – возвращает подстроку из строки, начиная с символа (**i**) длиной в один символ;

– Chr() – возвращает как текстовое знак в коде ANSI;

– Asc() – возвращает как целое код знака в кодировке ANSI;

– Trim() – удаляет из строки начальные и конечные пробелы.

Функции реализуют шифрование пароля методом замены. Заменяется каждый символ пароля. При замене учитывается позиция символа в строке. Текстовый элемент **txtParole** (пароль) – свободный элемент управления. Он предназначен для ввода пароля. Отличительной особенностью этого элемента является свойство «Маска ввода». Откройте список значений этого свойства и установите при помощи окна-построителя «Создание масок ввода» маску «Пароль». При работе с формой **Login** каждый символ пароля будет отображаться символом (*) звездочка. Последний этап – добавление в форму двух кнопок **Вход** и **Отказ**. Листинг 8 содержит код обработки события для кнопки **Вход**.

Листинг 8. Код события **Click** кнопки **Вход**

```
Private Sub Кнопка2_Click()
' Кнопка Вход
Dim Parole As String ' Пароль из базы данных
Dim ParoleEnter As String ' Введённый пароль
If IsNull([Forms]![Login]![ComboBox].Column(0)) Then
 ' Если фамилия не выбрана - Null
 ' Вернуть работу в форму
 MsgBox "Найдите свою фамилию в списке"_
 vbOKOnly + vbExclamation, "Внимание"
  Exit Sub
End If
' Значение пароля из таблицы tblUser
' Содержится в пятой колонке объекта ComboBox
' Нумерация колонок начинается с нуля
Parole = [Forms]![Login]![ComboBox].Column(4)
' Удаляем концевые пробелы
Parole = Trim(Parole)
If Len(Trim(Parole)) = 0 Then
 ' Если в таблице tblUSER был стерт пароль
 ' (Например через ODBC)
 SuperVisor = False  ' Идентификация не выполнена
 MsgBox "Пароль в таблице идентификации отсутствует." _
 & Chr(13) & "Операция сравнения паролей " & _
 "не может быть выполнена", _
 vbOKOnly + vbExclamation, "Внимание"
 Application.Quit acQuitSaveAll ' Выходим из приложения
Else
 If IsNull([Forms]![Login]![txtParole]) Then
 ' В поле введенного пароля - Null
 MsgBox "Вы забыли ввести пароль."_
  vbOKOnly + vbExclamation, "Внимание"
  ' Установка курсора в поле ввода пароля
```

```
    ' и возврат в форму
    [Forms]![Login]![txtParole].SetFocus
    Exit Sub
    End If
    ' Удаление концевых пробелов
    ParoleEnter = Trim([Forms]![Login]![txtParole])
    ' Зашифровка введённого пароля
    ParoleEnter = CrKod(ParoleEnter)
    ' Фамилия работника
    FAMILY = [Forms]![Login]![ComboBox].Column(0)
    If ParoleEnter = Parole Then
    SuperVisor = True  ' Идентификация выполнена
    ' Права доступа к пунктам меню и кнопкам форм
    ' Смена картинки главного окна программного комплекса
    ChangePicture = [Forms]![Login]![ComboBox].Column(6)
    ' Возможность изменять пароль
    ChangePassword = [Forms]![Login]![ComboBox].Column(7)
    MsgBox "Инспектор " & Trim(FAMILY) & " ! Доступ разрешён", _
    vbOKOnly + vbExclamation, "Результат идентификации"
    Else
    SuperVisor = False ' Идентификация не выполнена
    MsgBox "Инспектор " & Trim(FAMILY) & " ! Доступ запрещён", _
    vbOKOnly + vbExclamation, "Результат идентификации"
    Application.Quit acQuitSaveAll 'Выходим из приложения
    End If
    End If
    DoCmd.Close  ' Закрытие этой формы
    StartMainMenu ' Запуск главного меню
    End Sub
```

Если работник щёлкает кнопку **Вход**, не выбрав свою фамилию в поле со списком, то в нулевой колонке **ComboBox** в качестве фамилии содержится значение **Null**. Процедура прекращает обработку и возвращает управление в форму (**Exit Sub**). Для исключения возможности банального удаления пароля злоумышленником в таблице **tblUser** (заметим, что способов получения доступа к таблице MS Access достаточно) производится проверка длины пароля пользователя. Если пароль отсутствует, то работа программного комплекса прекращается. «Подсматривать» пароль через ODBC не имеет смысла, так как в таблице он зашифрован, а вводить его в форму требуется в раскодированном виде.

Список использованных источников

1 Наумов И. В. Ключевые процессы обеспечения информационной безопасности в системе защиты информации / И. В. Наумов, Р. А. Ешенко // Вестник ХГУЭП. 2017. № 3. С. 114–118.

2 Гурвиц Г. А. Microsoft® Access 2010. Разработка приложений на реальном примере / Г. А. Гурвиц. СПб. :

БХВ-Петербург, 2010. 496 с. (Профессиональное программирование).