How Do I Improve Security Using 4D for Flex?

Introduction

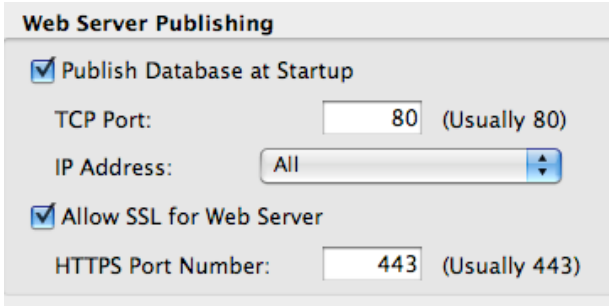
The .SWF file format used to deploy Flex and Flash application can be decompiled. Everybody can explore it. So, storing private data inside it, like password and user name, is not secure. Developers must be aware of this problem. This document shows you a way to improve security using 4D for Flex 1.0.

We are going to build a secure authentication system based on SSL, HTTP and Javascript (to communicate between the embedded .SWF and the HTML page).

We'll see finally that the next version of 4D for Flex (1.1) will provide a native secured authentication system that will greatly facilitate the process.

Prerequisites

SSL must be enabled on the 4D HTTP server.



1. HTML Form

First, we create an HTML Form for authentication. This form is submitted by POST to this URI: https://{host}/authUser

The user must call the form using an SSL connection (indicated in the URL by HTTPS).

2. 4D method, On Web Authentication

In the On Web Authentication 4D database method, we get the 'client IP address,' the 'User Name' and the 'User Password' parameters.

We check User Name and Password, generate a token, and save it with the client IP address in an array. (You could create the same session management system with storage in the database.)

After these operations, the 4D HTTP Server redirects the client to an HTML page containing the .SWF file with the token sent in the URL.

3. Pass the token to the .SWF file

We create a Javascript function called ‘getURLVars’ in the HTML file. This function, called by the SWF, returns the value for a specified variable. In this case, ‘id’.

```
<script type="text/javascript" charset="utf-8">

    //Get URL form variables and store them in an associative array
    function extractUrlParams(){
        var t = location.search.substring(1).split('&');
        var f = [];
        for (var i=0; i<t.length; i++){
            var x = t[ i ].split('=');
            f[x[0]]=x[1];
        }
        return f;
    }

    //Return the value for a specific variable submitted in the URL
    function getURLVars(varToGet){
        var getVars = extractUrlParams();
        return getVars[varToGet];
        //alert(varToGet) ;
    }

</script>
```

The URL is something similar to:
http://{host}/myAuthPage.html?id=45D18547DE4E236334R4R2380746G5

getURLVars function is called by the .SWF file to get the token through the Flash ExternalInterface object.
(http://livedocs.adobe.com/flex/3/html/help.html?content=19_External_Interface_04.html)

```
// Get ID with JavaScript
private function initID():void
{
        _resultSet = null;
        uidVar = ExternalInterface.call("getURLVars", "id");
        Alert.show(uidVar);
        fourDSQLService.connect();
}
```

The token is then used to replace the userName in the definition of the service. No other credentials are sent.

```
<fourD:SQLService id = "fourDSQLService"
    host = ""
    userName = "{uidVar}"
    userPassword=""
    autoConnect="false"
    result = "resultHandler(event)"
    fault = "faultHandler(event)"
    connect="connectHandler(event)"
/>
```

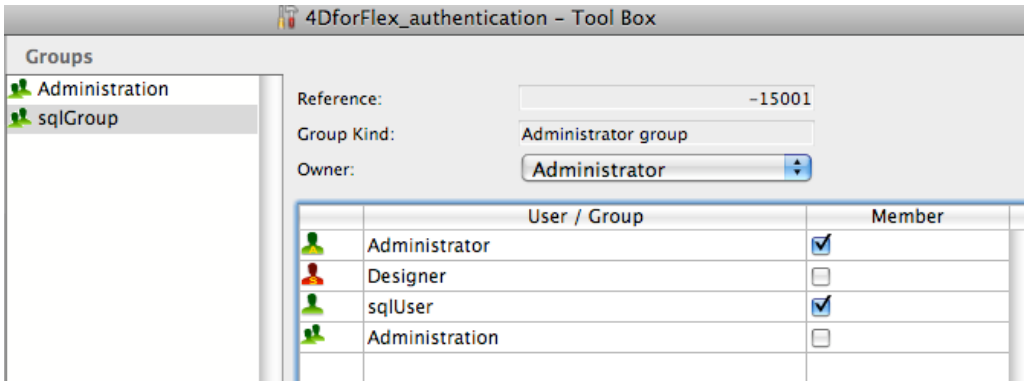
4. 4D method, On SQL Authentication

In the database method On Web Authentication, we test if the received token and the IP address match, and whether the session has not expired.
If everything is OK, we set the current user to a given account with limited rights.

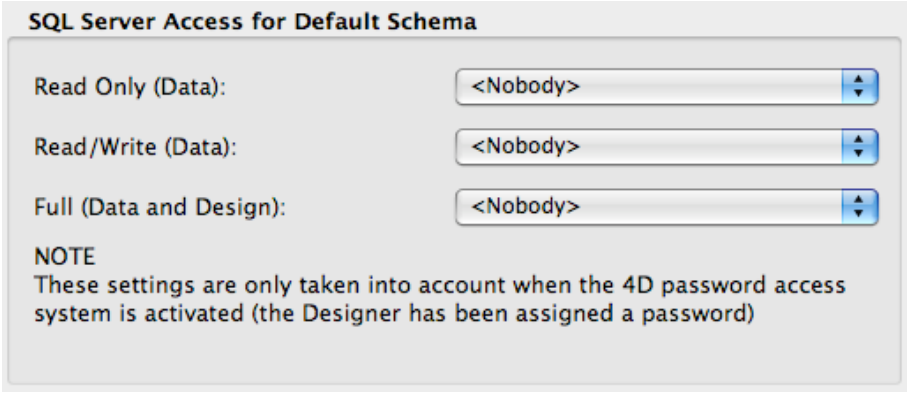
```
CHANGE CURRENT USER("sqlUser";"sql")
```

5. **Increase security with SQL Schemas**

If we stop here, the 4D for Flex client will get rights from user sqlUser who is member of sqlGroup.



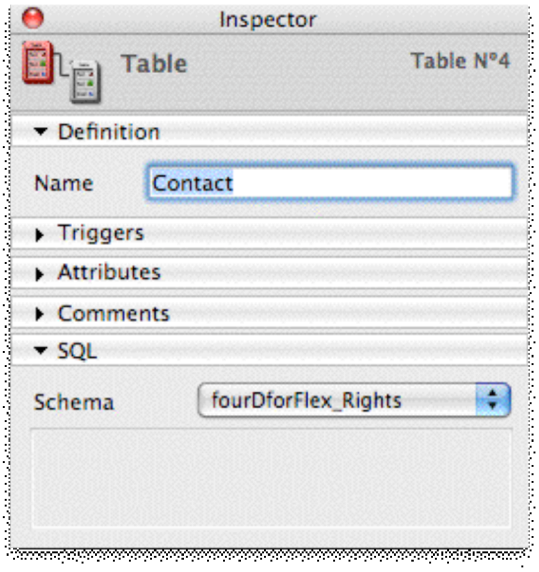
In 4D's Preferences we have declared Nobody to have read/write access to Default Schema.



6. Usage of SQL Schemas permits the definition of User Groups Rights

A good way to increase security is to add a SQL schema to restrict SQL access only to desired tables, for example, the table ‘Contact’.

```
Begin SQL
    CREATE SCHEMA fourDforFlex_Rights ;
End SQL
```



Afterward, we can change Groups Rights on Schemas in the database :

```
Begin SQL
    GRANT READ ON fourDforFlex_Rights TO sqlGroup
End SQL
```

Now, only the ‘sqlGroup’ has read rights for the ‘Contact’ table, and cannot read nor change other tables in the database.

Security considerations

Note: The communication protocol between the SWF and the 4D SQL Server is binary and not encrypted. That’s because Flash player doesn’t allow SSL on socket communication.

Note: A similar mechanism, using SSL authentication, will be implemented in 4D for Flex 1.1 in order to enhance security. It will provide a new class SQLSecuredService *allowing secured authentication.*

Example:

```
<fourD:SQLSecuredService id="_sqlService"

    httpPort="8080"
    httpUseSSL="false"

    showBusyCursor="true"
    prefetch="100"

    result="sqlResultHandler(event)"
    fault="sqlFaultHandler(event)"
```

```
connect="sqlConnectHandler(event)"
disconnect="sqlDisconnectHandler(event)" />
```

Conclusion

Security is a big challenge and a major concern when building Rich Internet Clients. We are very sensitive to this subject and will keep improving 4D for Flex throughout versions in this domain.