Mi-Forms Security Overview

1 Scope

This document is intended to provide technical information about the security and auditing capabilities of the Mi-Forms system as of version 10.5. The system is divided into four separate sections (designer, component, client, and server), each of which has its own security features. Where necessary, feature availability will be designated by Mi-Forms section.

2 Data Capture

During data capture, information relevant to the data element captured is also recorded. This information is available through the component's object model and is meant for long term data auditing purposes.

2.1 Session Data Element

Each time a session is opened in the Mi-Forms component, information about the environment on which it was opened is recorded. Each record includes username, device name, session open time, Mi-Forms versioning, operating system versioning, memory availability, battery level, session end time and potentially session transmittal time. These pieces of information are embedded directly into the session's XML data format as shown in figure 1 below:

```
<SESSION_DATA START_TIME="1397493960" END_TIME="1397493963" TRANSMIT_TIME="1397493963"
SESSION_ID="C5DEMO1_633413545333125000" NUM_RETRIES="0" SYNC="YES">
  <CREDENTIALS USERNAME="MicoDemo" DEVICE_ID="C5DEMO1" />
  <DEVICE_STATE OS="Win32NT" OS_VERSION="6.2.9200.0" MI-FORMS_VERSION="10.0.0.0 (10.0.0.0)"
TOTAL_MEMORY="1038320" FREE_MEMORY="478964" BATTERY_LEVEL="96" />
</SESSION_DATA>
```

Figure 1 – Session data element XML

A given session may have an unlimited number of session data elements recorded depending on the number of times the data record has been opened and processed. Each session data element is accessible via the Mi-Forms component's object model and is retained for the life of the session.

Note that device information availability varies by device, and not all platforms may support all metrics.

2.2 Ink Data Recording

When ink data is captured in a Mi-Forms session, information additional to coordinate data is recorded. Each instance of ink's user is recorded, each stroke's timestamp is recorded, and each point's pressure data is recorded if available.

As an example, the letter "A" is written on a Windows TabletPC and shown in figure 2 below.

Figure 2 – Image of the letter “A” written on a Windows TabletPC

The data recorded for this ink follows in figure 3.

```
<INK_DATA_XML TIMESTAMP="1210620678" USER="MicoDemo" ENTRY_MODE="INK">
 <ink>
  <captureDevice id="Mi-Forms Ink Capture">
   <channelList>
    <channelDef name="X">
     <representation type="integer" />
     <resolution value="0.001" units="cm" />
    </channelDef>
    <channelDef name="Y">
     <representation type="integer" />
     <resolution value="0.001" units="cm" />
    </channelDef>
    <channelDef name="Pressure">
     <representation type="integer" />
     <resolution value="1" units="units" />
    </channelDef>
   </channelList>
  </captureDevice>
  <traceFormat id="Mi-Forms Trace" version="1.0">
   <regularChannels>
    <channel name="X" type="integer" mapping="*" />
    <channel name="Y" type="integer" mapping="*" />
    <channel name="Pressure" type="integer" mapping="*" />
   </regularChannels>
  </traceFormat>
  <brush id="1" color="0" width="53" transparency="0" />
  <trace id="1" startTime="1210620678" brushRef="1">2305 2729 34 2305 2729 53 2305 2729 70 2305
2729 85 2305 2729 98 2305 2729 109 2305 2729 119 2307 2708 129 2307 2708 138 2311 2679 146 2325
2666 155 2327 2635 164 2341 2607 173 2343 2563 181 2361 2527 189 2371 2474 195 2394 2418 201
2409 2352 207 2430 2288 211 2453 2222 215 2478 2150 218 2502 2076 219 2527 2007 219 2550 1944
221 2573 1878 221 2598 1820 220 2619 1767 218 2635 1716 219 2654 1670 218 2671 1632 218 2686
1594 218 2702 1560 217 2715 1534 216 2727 1509 216 2738 1490 216 2749 1471 216 2749 1471 216
2765 1466 216 2765 1466 216 2778 1446 215 2778 1446 214 2786 1474 215 2786 1474 215 2799 1509
214 2802 1541 214 2807 1585 214 2813 1652 213 2818 1735 213 2823 1821 215 2826 1915 214 2831
2031 216 2835 2156 215 2838 2273 215 2842 2387 214 2843 2500 216 2846 2603 214 2849 2686 212
2855 2763 210 2857 2821 208 2861 2865 206 2866 2892 204 2873 2915 199 2873 2915 192 2882 2936
176 2882 2936 144 2882 2936 107 2882 2936 68 2882 2908 32 </trace>
  <trace id="2" startTime="1210620678" brushRef="1">2410 2276 43 2410 2276 77 2410 2276 111
2433 2258 139 2459 2257 163 2491 2239 182 2545 2229 195 2597 2205 205 2667 2194 209 2741 2185
189 2818 2183 153 2886 2188 113 2945 2197 71 2989 2202 32 </trace>
 </ink>
```

```
</INK_DATA_XML>
```

Figure 3 – Data recorded for the "A" written on a TabletPC

Note that the A consists of two strokes (traces) in the XML data, each of which is time stamped and each of which contains X, Y, and pressure data. This audit data is available via the Mi-Form component's object model and is retained for the life of the session.

Note that not all platforms support pressure information. On these platforms, pressure will always be listed as "255".

2.3 Textual Data Recording

When a field's value (text data) is updated, the data that used to be in the field is not replaced, instead it is moved down a list of available data elements, and its most current data is considered its new value. Every time a data element is recorded with a field, its timestamp and updating user is recoded as well.

As an example, the constrained text field in figure 4 contains three boxes. In this example, the field's data was initially recorded as "ABC" by User1 and was then later updated to "XYZ" by User2. Figure 4 shows that its current value reflects the "XYZ" data.

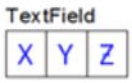

Figure 4 – Constrained text field with value "XYZ"

Figure 5 below illustrates the data that is captured in the session's XML data. Be aware that for the purposes of this example, the ink recorded as well as alternate recognition guesses have been discarded.

```
  <CTEXT_GROUP NAME="TextField" POSITION="2.5,4.55" SIZE="2.34,0.98" TAB_ORDER="1"
CAPTION="TextField" FORMATTYPE="TEXT" APPENDCUR="NO" DECPLACES="0" LEXICON="NONE">
    <DATA TIMESTAMP="1210623102" RECO_CONFIDENCE="0.00" USER="User1"
ENTRY_MODE="INK">ABC</DATA>
    <DATA TIMESTAMP="1210623111" RECO_CONFIDENCE="0.00" USER="User2"
ENTRY_MODE="INK">XYZ</DATA>
    <CTEXT NAME="TextField_001" POSITION="2.5,4.55" SIZE="0.78,0.97" SEQ_NUM="1" TYPE="ALPHA">
     <DATA TIMESTAMP="1210623102" RECO_ENGINE="Mi-Co Recognition Engine -- MS"
RECO_CONFIDENCE="93.38" USER="User1" ENTRY_MODE="INK">A</DATA>
     <DATA TIMESTAMP="1210623111" RECO_ENGINE="Mi-Co Recognition Engine -- MS"
RECO_CONFIDENCE="99.62" USER="User2" ENTRY_MODE="INK">X</DATA>
    </CTEXT>
    <CTEXT NAME="TextField_002" POSITION="3.28,4.55" SIZE="0.78,0.97" SEQ_NUM="2"
TYPE="ALPHA">
     <DATA TIMESTAMP="1210623102" RECO_ENGINE="Mi-Co Recognition Engine -- MS"
RECO_CONFIDENCE="98.64" USER="User1" ENTRY_MODE="INK">B</DATA>
```

```
    <DATA TIMESTAMP="1210623111" RECO_ENGINE="Mi-Co Recognition Engine -- MS"
RECO_CONFIDENCE="99.82" USER="User2" ENTRY_MODE="INK">Y</DATA>
   </CTEXT>
   <CTEXT NAME="TextField_003" POSITION="4.06,4.55" SIZE="0.78,0.97" SEQ_NUM="3"
TYPE="ALPHA">
    <DATA TIMESTAMP="1210623102" RECO_ENGINE="Mi-Co Recognition Engine -- MS"
RECO_CONFIDENCE="99.85" USER="User1" ENTRY_MODE="INK">C</DATA>
    <DATA TIMESTAMP="1210623111" RECO_ENGINE="Mi-Co Recognition Engine -- MS"
RECO_CONFIDENCE="99.66" USER="User2" ENTRY_MODE="INK">Z</DATA>
   </CTEXT>
  </CTEXT_GROUP>
```

Figure 5 – XML representation of the data recorded for the field

Note that data is recorded both on the group level as well as the individual box level and that data is not discarded when the field's value changes. Each data element of the group and each box is available via the Mi-Forms component's object model and is retained for the life of the session.

Data elements as above are kept for all textual value recording field types such as constrained text, freeform, checkbox, and picklist. Audit trails may be disabled per field by the forms' designer if it is not necessary.

2.4 Additional Data Recording

Other data recording such as pictures recorded in image annotations also record user and time information. In the case of devices that support it, images may also be geotagged automatically by the operating system.

2.5 Session Hashing

When the Mi-Forms component exports data in its standard XML format, a security hash is applied that is computed against the entirety of the XML document as well as an internal password. An example of such a security hash follows in figure 6:

```
<SECURITY_HASH HASH_CODE="66-50-36-A1-1A-64-13-DA-39-1B-99-C6-32-18-93-A2-3D-5A-7F-8F" />
```

Figure 6 – Example security hash for a Mi-Forms session

At session load time, the security hash is compared against what the computed hash code for the loaded session should be. If they match, the session is loaded and its security state is set to "Valid". If they do not match, the session is loaded, but its security status is set to "Invalid". If the loaded session does not have a security hash (e.g. it was maliciously removed), the session is loaded, but its security status is set to "Unverified".

The Mi-Forms Windows Client will by default warn users when sessions being loaded are in the Unverified or Invalid security state. It is at the discretion of the user as to whether the session should be opened at that point.

3 Data Storage

The default storage mechanism for a session data record is a native format XML file. This XML file may be loaded into the Mi-Forms component directly as text. There is no direct encryption of this format at the component level, but applications built around the component typically employ encryption of their data stores.

3.1 Client Data Storage

3.1.1 Windows Data Storage

The Mi-Forms Windows Client automatically encrypts all form templates and sessions using 64bit DES technology. Mi-Co recommends that full disk encryption be used in conjunction with the Mi-Forms Windows Client as needed.

Note: This encryption is not enabled in the Basic licensing level of Mi-Forms.

3.1.2 iOS Data Storage

The iOS Client uses iOS's data protection which supplements devices' basic hardware encryption layer by protecting hardware encryption keys with a user's passcode. It provides a layer of protection for data captured and stored locally on the device.

3.1.3 Android Data Storage

The Android Client uses Android's Security Setting Device Encryption feature to provide encryption of all data related to the app by the user's passcode.

3.2 Server Data Storage

The Mi-Forms Server automatically encrypts all form templates, sessions, and rendered images using 256 bit AES technology.

4 Authentication

While the Mi-Forms component itself does not enforce authentication, its session format provides a framework upon which authentication can be built. Firstly, as shown in the data capture section above, each ink and data element is stamped with a username. This username must be provided by the component's wrapping application. Additionally, two form level properties AuthenticateOnStart and AuthenticateOnFinish can be set to require the wrapping application to authenticate at given times.

4.1 Client Authentication

The Mi-Forms Client allows for username and password combinations, verified via a Mi-Forms Server (see below). Once a user has successfully authenticated once on the Client, the user may continue to use those credentials until they are known to be different (e.g. from a server authentication).

The client prompts the user on application start before displaying an available list of form templates and saved form sessions. This is illustrated in figure 7 below in the Windows Client, but the process is similar for other platforms:

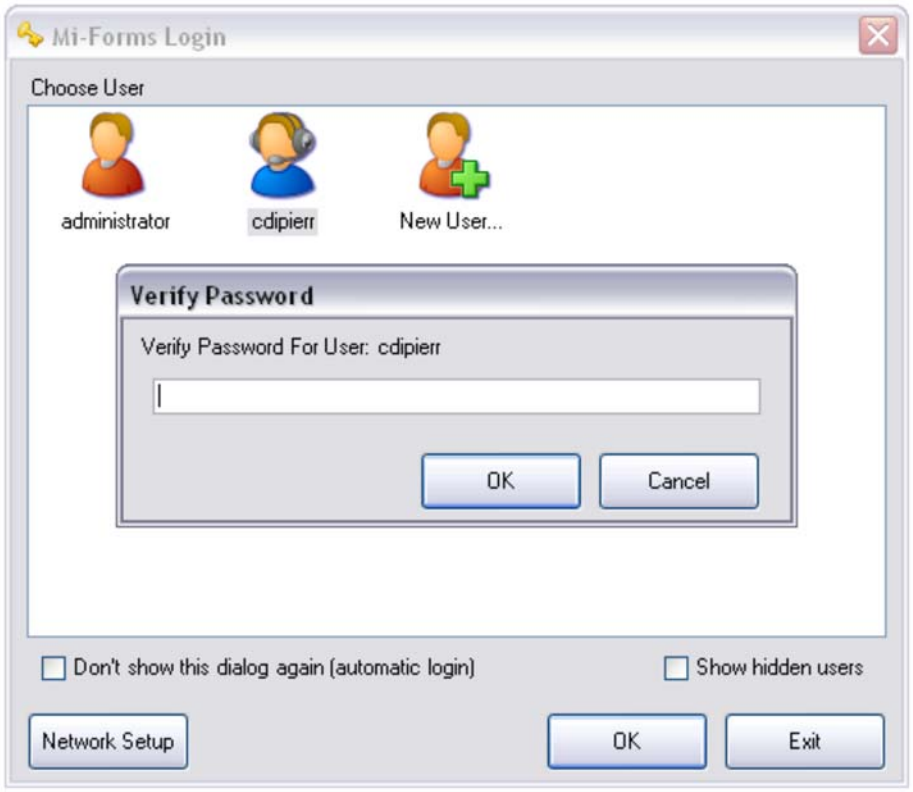

Figure 7 – Authentication on application launch of Mi-Forms Windows Client

Additionally, if a form has been designed to require authentication on start or finish, the client prompts the user to re-enter their password at the appropriate time. Figure 8 below shows an example of a form being authenticated on finish:

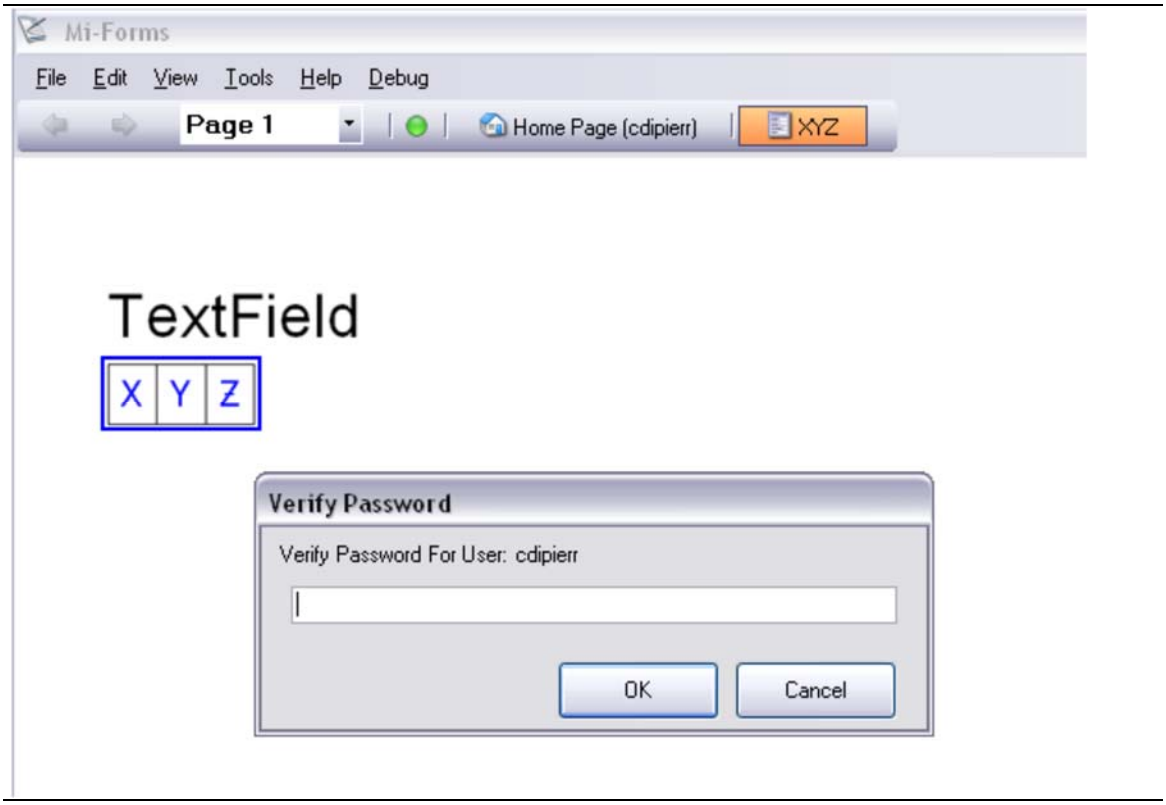

Figure 8 – Authentication on form finish in Mi-Forms Windows Client

Failure to authenticate correctly at start will result in the form not being displayed. Failure to authenticate correctly on finish will result in the form's data not being exported.

4.2 Server Authentication

The server has the ability to be the master arbiter of usernames and passwords that can be used by a form filling application such as the client.

The server allows an administrator to specify password policies for all users including providing three levels of password strength and expiration. Additionally, it allows account lockout settings based on failed authentication attempts. These password and user security settings are compliant with 21 CFR Part 11. Figure 9 shows the server's UI for managing these settings:

Figure 9 – Server UI for specifying password quality settings

Figure 10 below show example password strength settings. Note that each password strength is enforced by a regular expression that may be modified as necessary by an administrator:

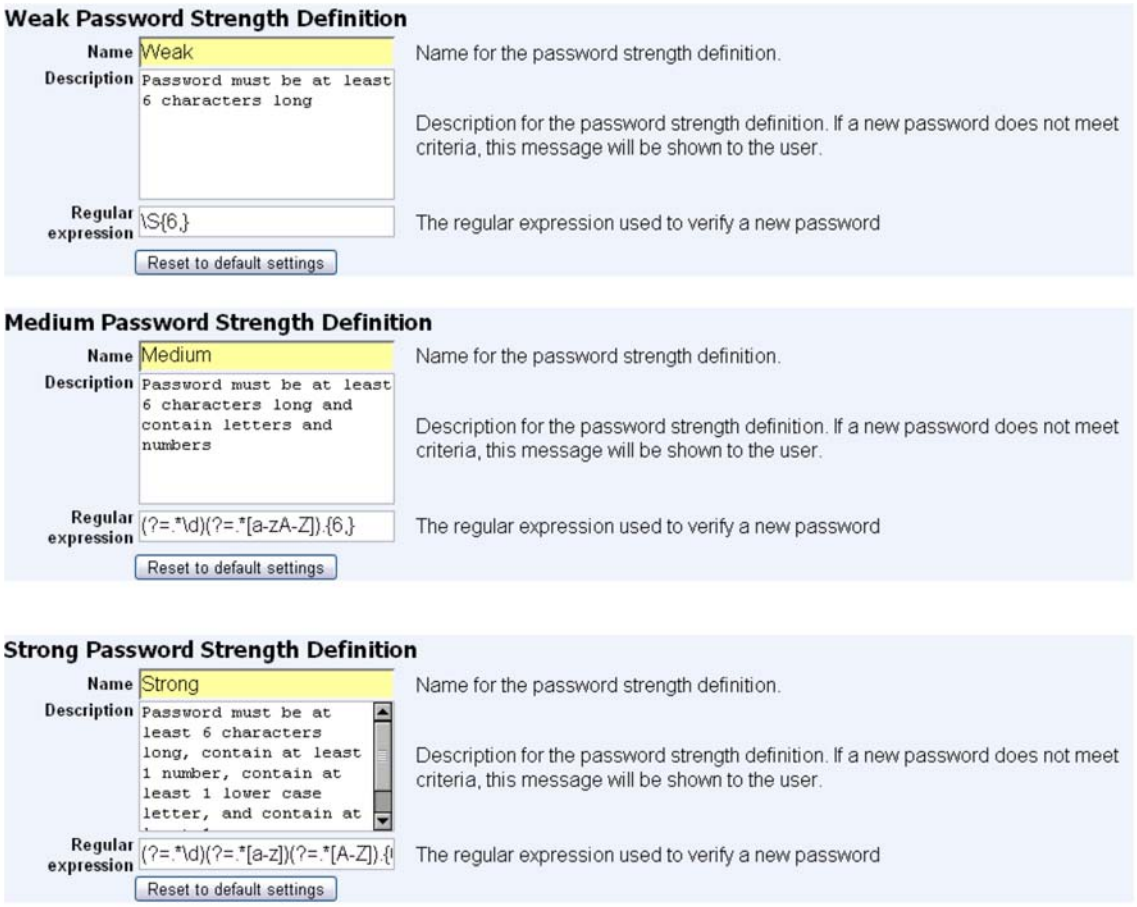

Figure 10 – Server UI for specifying password strengths

While the UI shown above specifies default security settings, individual user security settings may also be set as needed. Figure 11 below shows that a user's account may be locked until a specified date and that they may be required to change their password upon next login:

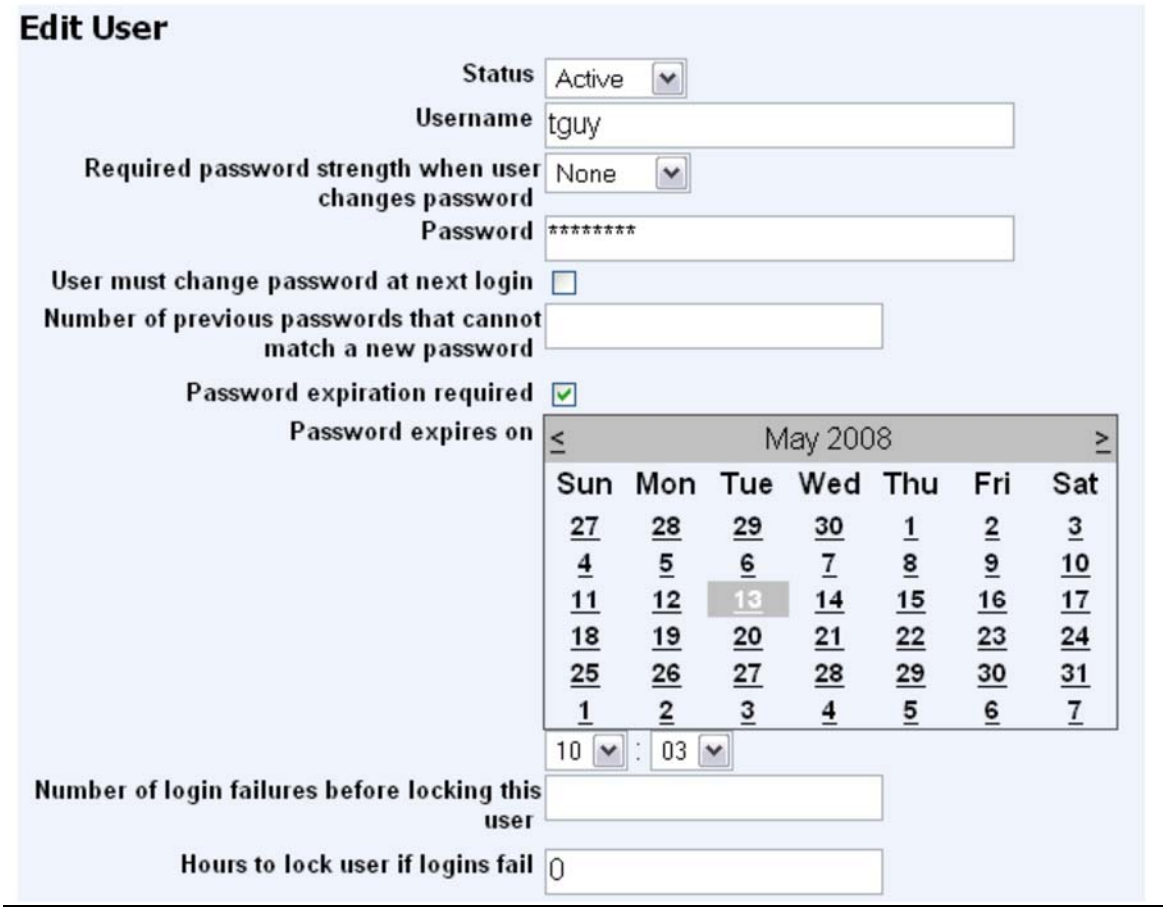

Figure 10 – Server UI for specifying individual user security settings

4.2.1 Server Active Directory Authentication

It is possible to configure the Mi-Forms Server to use an Active Directory domain controller as its authentication source. If this is done, the domain controller specifies all security policies and they are not modifiable via the Mi-Forms Server itself.

Note: Active Directory authentication is only available in the Enterprise licensing level.

4.2.2 Privileges

The Mi-Forms Server provides three levels of privileges to known users. These privileges are assigned on the group level and users inherit privileges from all groups of which they are a member. The privileges are as follows:

User – Allows a user to download form templates assigned to their group(s) as well as upload sessions created from these templates. Also allows the user to see sessions created by anyone in any group of which they are a member.

Form Filler – Allows a user to fill forms via the server's web interface, including access to form sessions in that user's queue or queues belonging to groups of which the user is a member.

Template Filler – Allows a user to fill forms via the server's web interface, but only allows access to new form templates, not to form sessions.

Publisher – Allows a user to publish new form templates

Administrator – Encompasses all privileges in User and Publisher levels and provides additional functionality. Users with this privilege have the ability to configure security settings, client updates, and licenses updates. They may also manage user permissions and group memberships. They may see all sessions uploaded to a given customer as well as perform actions to them such as assigning them to different queues, deactivating/reactivating them and forcing an unlock.

5 Communication

The Mi-Forms Server provides a web service interface through which client side applications such as the Mi-Forms Client may communicate. Each web service request requires authentication via established usernames and passwords already known to the server. While no encryption is performed by the client application or server, it is recommended that production servers be configured to accept SSL (HTTPS) connections only. Doing so will automatically encrypt all template, session, and other data transferred to and from the Mi-Forms Server.

5.1 Mi-Forms App for iOS

The Mi-Forms App for iOS is configured such that if server settings are entered with a requirement for a secure connection (SSL is checked), the Mi-Forms Server web site must provide a valid SSL certificate from a CA Root provider. Self-signed certificates are not considered valid certificates by iOS.