General Security Guidelines

The following methods are best practices for ensuring the security of Agiloft and the server it runs on. We highly recommend that you use each method below to protect your data and reduce the likelihood of security vulnerabilities in your system.

Follow the Principle of Least Privilege

Do not allow users to have privileges they do not need or do not have the skills to use safely. For example, a user with the ability to delete all records in a table in one operation can do considerable unintentional damage if they are not familiar enough with Agiloft's architecture. Only trusted and trained users should be placed in the Admin group because members of this group can make drastic changes to the structure and data of your system.

Use Secure Sockets Layer (SSL)

Use SSL via HTTPS to secure web browser connections to the Agiloft server. Using standard HTTP to connect to the Agiloft server exposes passwords and potentially sensitive information to anyone able to monitor network traffic. The ability to intercept network traffic also opens up additional methods of attack.

To connect to your web server using SSL, you need to configure it manually because SSL is not configured by default. This involves purchasing or generating a server certificate that authenticates your server to the clients. This configuration differs depending on the host operating system and the web server software you use. The following resources may help:

- Securing Your Apache 2 Server with SSL
- Van's Apache SSL/TLS mini-HOWTO
- How to implement SSL in IIS

Even if you must allow access to some accounts through standard HTTP, ensure that HTTPS is used to access more sensitive accounts, such as those in the Admin group or with login access to the admin console.

Restrict Login Access to the Agiloft Server

Make sure that only trusted and experienced users have access to the Agiloft server. A root user on Unix/Linux or a user in the Administrators group in Windows can use their special privileges to circumvent Agiloft internal security. Even unprivileged users can circumvent security by using local web access, so make sure you appropriately restrict all methods of accessing the server.

Restrict Services Accessible on the Agiloft Server

Treat the Agiloft server as you would any other sensitive server. Only allow connections essential for Agiloft operation, such as HTTP and HTTPS, and connections required for administration, such as SSH for Unix/Linux or Terminal Services for Windows. Additional services or applications that run on the same server machine, including other web applications, may potentially contain security holes that could lead to the compromise of Agiloft data.

The default services installed with most recent Linux distributions are generally minimal. To aid security, use the nmap tool to verify which ports are exposed on your server, which may return a result like the following:

```
linux# nmap -sS wizard.example.com
Starting Nmap 4.00 ( http://www.insecure.org/nmap/ ) at 2006-12-14 18:12 PST
Interesting ports on wizard.example.com (10.0.0.1):
(The 1667 ports scanned but not shown below are in state: filtered)
PORT      STATE  SERVICE
22/tcp    open   ssh
80/tcp    open   http
113/tcp   closed auth
443/tcp   open   https
8080/tcp  open   http-proxy
MAC Address: 00:E0:81:00:00:12 (Tyan Computer)
Nmap finished: 1 IP address (1 host up) scanned in 64.320 seconds
linux#
```

For reference, these are the TCP ports normally used by Agiloft:

Port number	Description
80	The standard HTTP port that connects to the Apache or IIS web server. The `/gui2/` URL is forwarded to the Tomcat server and is the normal unsecured access port to the Agiloft application.
8080	The native connection port to the Tomcat server that is part of the Java framework behind Agiloft.
443	The standard HTTPS port for web service over SSL. This is either forwarded to the Tomcat server by the native web server or forwarded directly to port 8443 by Linux kernel using the internal firewall module.
8443	The native HTTPS port that Tomcat may be configured to listen to. It is often better to use the SSL engine in Tomcat with requests forwarded from port 443 than to configure the native Web server for SSL and request forwarding.
3306	The standard server port for MySQL, the default Linux back-end database, This port is not exposed externally; in other words, it is bound only to `localhost`.