Data Transfer Service: SFTP

Preface

Customers can utilize this service to deliver data files to Materialogic's system or retrieve data files from Materialogic's system. This document will provide a guideline for securely exchanging information with Materialogic.

Audience and Prerequisites

This document is intended for technical personnel responsible to construct and configure a SFTP communication channel with Materialogic. These individuals will require administrative access and skills over the customer's system and network.

Overview

Materialogic offers a communication channel and server for the exchange of data through Secure FTP. When requested by customer, Materialogic will configure its SFTP server to allow customer access.

Customer Setup

Materialogic recognizes that customer data security is of utmost importance. Therefore, customers must consider the following elements of security.

Firewall

At the firewall level, customers are required to register their IP address(es). They can provide this in the form of either an IP address (e.g., 208.208.208.208) or an IP address range (e.g., 208.208.208.0/28). The firewall rules will then be set to only allow connections from the identified system through standard port (TCP 22). Once the IP address or IP address range are supplied, the firewall will be configured to allow access to the SFTP server.

SFTP

At the time of the firewall configuration, the home directory and username/password will be setup. The host name for the SFTP server is ftp.materialogic.com. Again, the SFTP server is limited at the firewall level. Therefore, it is imperative the firewall rules be updated at the same time the account is created.

Directory Structure

Once the account access is configured, customers can transfer files to and from Materialogic.All customers are jailed to their accounts. Each account's root directory on the Materialogic server will contain at least three subdirectories.

from_ml

This directory will contain files to be retrieved by customer. After the files are retrieved, the customer must delete the file from this directory.

Materialogic

Materialogic's security procedures will require removal of files after one business day. An automatic scanning program will monitor customer's compliance to this security rule and will automatically remove the old files that have not been deleted by the customer.

to_ml

This directory will contain files delivered by the customer, which are to be used by Materialogic. Placed files will only remain in this directory for a short time and until the relocation program recognizes their placement. Once their placement is recognized, the files are relocated to an internal server and removed from this directory.

.ssh

This directory will contain SSH authorized keys used in negotiating a connection to the SFTP server. SSH keys and the contents of this directory are discussed in the next section.

SSH Keys

In addition to conventional username/password authentication, SSH supports public key authentication. **Materialogic ***strongly*** prefers key based authentication.** It's more secure, more reliable, and easier to automate.

Key Types

SSH supports two types of keys: Digital Signature Algorithm (DSA) and Rivest-Shamir-Adleman (RSA). **Materialogic ***strongly*** prefers RSA over DSA because of weaknesses in DSA's random *k*-selection.** Therefore, DSA will be supported for existing keys, but any new keys created **MUST** be RSA. The other consideration about the keys is the key size, in bits. Standard key sizes for DSA are 1024, 2048, and 3072. Key sizes for RSA are 1024, 2048, and 4096. RSA 4096 is the most secure, and doesn't require any more effort than the weaker size keys do. Therefore, RSA 4096 bit keys are the preferred key of communicating with Materialogic.

Key File Formats

There are two major file formats for public keys: X509 (PEM) format and the OpenSSH format. Materialogic prefers the OpenSSH format where possible. Both files can be viewed in Notepad or other text editor. Here are examples of both formats:

OpenSSH Format

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAgEAlFp/MEyihjkyypfB67emDI4YtNjwnoA9/eu1LVRpPsOem5Bu1x6VesR6
ER0XObMpASoVtP6ASbxfdnqCLImYKCc2Qf0UDGUNVSueZbSkKE/j/ZXhJijURUBLGs86Mf2j592mHT5+eaG2
edDyBYguffWiJgtIdoAyAJkgQ0rjRaUmgvJ4IrjtcJ3cGLiBomiaQ9fR6M8dUiC4l+KNgJOOFaO1Oohm3BGT
CpNJK2RYMDutH2uJofZ+xbZd73qZax0X31DTxKDIxzkUNe2H8A85RszymBAjDu7E0ILeH64acbqU1r6vBC9C
qdu+ooCtHi+WZLSaX6Cq4c3FS5VcgMKHvfBmywBNP3ZAIq4lt6VGwdN2CxWL2Aw7HyQ2AUGBPFIJiCdfXK/L
AwDyE65xYeLjPX/cBF2Jsa/amCacuJbNGHQDvcN/4dAW/aaxIy2TCWLGDGwx+d1RLReq/wH48ZNIDVBhcGnf
Td4P/zAyo5PuqzUzdQWdfBVi3CPBG9ShAj8/yVIFBq8e1XQ1UHrN9ba6ALP8q2ibLa08dblEbVrJkq0DvK0u
r0FfZEwGyxzYeqxtuNIrc7VovkXBItvGphsC/f4MJFEknhCWJlyQJ15+vXUgCgnLN67LD640vmfvfuPh4g86
0Q8EdNwwtUDAQtwak2OamGbKmJ5k/hHpXnycW28= user@mlrnd02
```

X509 Format

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted from OpenSSH by user@mlrnd02"
AAAAB3NzaC1yc2EAAAABIwAAAgEAlFp/MEyihjkyypfB67emDI4YtNjwnoA9/eu1LVRpPs
Oem5Bu1x6VesR6ER0XObMpASoVtP6ASbxfdnqCLImYKCc2Qf0UDGUNVSueZbSkKE/j/ZXh
JijURUBLGs86Mf2j592mHT5+eaG2edDyBYguffWiJgtIdoAyAJkgQ0rjRaUmgvJ4IrjtcJ
3cGLiBomiaQ9fR6M8dUiC4l+KNgJOOFaO1Oohm3BGTCpNJK2RYMDutH2uJofZ+xbZd73qZ
ax0X31DTxKDIxzkUNe2H8A85RszymBAjDu7E0ILeH64acbqU1r6vBC9Cqdu+ooCtHi+WZL
SaX6Cq4c3FS5VcgMKHvfBmywBNP3ZAIq4lt6VGwdN2CxWL2Aw7HyQ2AUGBPFIJiCdfXK/L
AwDyE65xYeLjPX/cBF2Jsa/amCacuJbNGHQDvcN/4dAW/aaxIy2TCWLGDGwx+d1RLReq/w
H48ZNIDVBhcGnfTd4P/zAyo5PuqzUzdQWdfBVi3CPBG9ShAj8/yVIFBq8e1XQ1UHrN9ba6
ALP8q2ibLa08dblEbVrJkq0DvK0ur0FfZEwGyxzYeqxtuNIrc7VovkXBItvGphsC/f4MJF
EknhCWJlyQJ15+vXUgCgnLN67LD640vmfvfuPh4g860Q8EdNwwtUDAQtwak2OamGbKmJ5k
/hHpXnycW28=
---- END SSH2 PUBLIC KEY ----
```

To convert a X509 file to an OpenSSH file, run:

```
ssh-keygen -i -f x509-file > OpenSSH-file
```

Authorizing Keys

Once you have the key in an OpenSSH format, the key needs to be saved (or appended) to the `authorized_keys` file, which is a simply a file containing the keys authorized to use this account, one per line. Either use a text editor, or append to the file:

```
cat OpenSSH-file >> /home/user/.ssh/authorized_keys
```

Once this is done, permissions need to be verified. OpenSSH is very strict on the permissions of `.ssh/*` and will refuse to use keys with insecure permissions. Make sure the `.ssh` directory is `chmod`'ed to 700 and all the files within that directory are `chmod`'ed to 600.

Generating Keys

Generating keys will be limited for internal testing purposes only. Because of the sensitivity of private keys, clients must generate their own private keys. ***Materialogic will NEVER generate SSH keys on the client's behalf.*** To generate a new key, use the `ssh-keygen` command:

SSH Keygen Conversation

```
$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
41:b6:4a:7b:89:53:77:da:4a:91:db:33:89:bb:77:cf user@mlrnd02
```

That will generate two files: `id_rsa` containing the private key and `id_rsa.pub` containing the public key. To use the key pair, authorize the key for the account(s) it is to be used for by following the instructions above.

Sending Keys

Sending keys raises security concerns and must be done carefully. Careless actions can result in the compromising of the keys.

Public Keys

Public keys are, by definition, public files and are safe to send by conventional methods such as Email, JIRA, etc.

Private Keys

Simple: don't. ***Private keys are NEVER to be sent outside Materialogic.*** The security concerns of sending private keys are too substantial to allow the sending or private keys. This is why Materialogic will never generate keys on client's behalf.