LISTSERV Maestro Admin Tech Doc 8

Restricting Access

October 20, 2022 | © L-Soft Sweden AB
lsoft.com

This document is a LISTSERV Maestro Admin Tech Doc. Each admin tech doc documents a certain facet of the LISTERV Maestro administration on a technical level. This document is number 8 of the collection of admin tech docs and explains the topic “Restricting Access”.

Last updated for LISTSERV Maestro 10.1-9 on October 20, 2022. The information in this document also applies to later LISTSERV Maestro versions, unless a newer version of the document supersedes it.

All of L-Soft's manuals are also available at: **http://www.lsoft.com/manuals.html**

L-Soft invites comment on its manuals. Please feel free to send your comments by e-mail to: MANUALS@LSOFT.COM

Table of Contents

1 Password Protection

Access to LISTSERV Maestro is protected with login passwords for different types of users: LISTSERV Maestro users, LISTSERV Maestro administrators and list subscribers.

Note: LISTSERV Maestro does not store passwords, but only password hashes, using the very secure BCrypt password hashing algorithm[1] (with cost factor 12).

Note: When entering the password on the LISTSERV Maestro or subscriber area login page, this password is then transferred from the user's browser to the LISTSERV Maestro server. If LISTSERV Maestro uses the unencrypted standard HTTP protocol, then the password is transferred in unencrypted plain text and can potentially be read by 3rd parties. It is therefore recommended to secure access to LISTSERV Maestro with HTTPS (see LMA Admin Tech Doc 9), so that all communication between the user's browser and LISTSERV Maestro, including the password, is securely encrypted.

1.1 Recover Access After Forgetting The Password

Because LISTSERV Maestro does not store the actual passwords, it is not possible to recover a forgotten password. It is however possible to set a new password, using the following options:

- **For standard LISTSERV Maestro users**

  Option 1: Use the "*Forgot Password?*" link on the LISTSERV Maestro login page to set a new password. This requires that the account has an account email address configured and can receive the confirmation email at this address. If this is not the case, use the second option instead.

  Option 2: Use an administrator account to login to LISTSERV Maestro and go to the Administration Hub (HUB). Go to **Main Menu → Accounts and Identities**, select the **User Accounts** tab, then select the user account in question, then select **Main Menu → Account Settings → User Authentication**.

- **For the default "admin" account**

  On the server where the Administration Hub (HUB) component is installed, delete the following file manually:

  `[maestro_folder]/hub/accountreg/account.admin&password_digest.s`

  This resets the password of the default "admin" account to the default password "*admin*". This default password must be changed by the administrator upon the next login.

- **For another administrator account** (other than the default "admin"):

  Use the default "admin" account (or another different administrator account) to login to LISTSERV Maestro and go to the Administration Hub (HUB). Go to **Main Menu → Accounts and Identities**, select the **Administrator Accounts** tab, then select the administrator account in question, then select **Main Menu → User Authentication**.

[1] BCrypt is used since LMA 7.2-1. Previously, LISTSERV Maestro used salted SHA-256 hashes. Existing hashes initially remain in the previous hash format, until the user or subscriber logs in the first time after the upgrade to LMA 7.2-1 (or later). As part of this first login, the password is rehashed with the new BCrypt algorithm.
Note: If a stand-alone list or list group has the "LISTSERV Interface Link" option enabled, then for subscribers in this list or group, the passwords are also stored by the LISTSERV component itself, which in the current version uses SHA-256 hashes.

- **For List Subscribers**

  Use the "*You don't have the password? Define it here (requires email confirmation)*" link on the subscriber area's login page to set a new password. This requires that the subscriber can received the confirmation email at his registered subscriber email address.

  If this is not possible (for example if the email address is "dead" and the whole reason why the subscriber wants to login is because he wants to change the address, only he also forgot the password), then contact the owner of the subscriber area or subscriber list and ask for the subscriber address to be changed to an address on which the subscriber *can* receive emails.

  Once the address has been changed, use the link as described above to set a new password for that address.

1.2 Password Quality

A good password is a password with high entropy. The higher the entropy, the more difficult it is to break the password hash. The simplest way to increase the entropy in a password, and thus improve the password's quality, is to make the password longer and/or use a wide range of different characters.

LISTSERV Maestro has two features that can be used to motivate (or force) users to use passwords with better quality, i.e. with higher entropy: Enforce a minimum password length and/or enforce a certain minimum password complexity.

1.2.1 Minimum Password Length

In a fresh LISTSERV Maestro installation, passwords must be at least 10 characters long. You can set your own minimum password length (but no shorter than 6) as the administrator in the Administration Hub (HUB): Go to **Main Menu → Security Options**.

Note: This setting applies only to LISTSERV Maestro accounts and administrators. It does not apply to subscriber accounts. For the subscriber accounts, the minimum password length is always six characters.

1.2.2 Password Complexity

A fresh LISTSERV Maestro installation has the "Password Complexity" option enabled. With this option enabled, LISTSERV Maestro only accepts passwords that contain at least one digit, one upper-case letter, one lower-case letter, and one special character that does not fall into the three other categories.

You can set this option as the administrator in the Administration Hub (HUB): Go to **Main Menu → Security Options**.

Note: This setting applies only to LISTSERV Maestro accounts and administrators. It does not apply to subscriber accounts. For the subscriber accounts, no complexity rules are enforced.

1.3 Automatic Password Expiration

In a fresh LISTSERV Maestro installation, passwords never expire, i.e. once set, a user can login with this same password forever.

To force users to change their passwords on a regular schedule, LISTSERV Maestro allows you to define a "Maximum Password Age in Days". With this option enabled, each password is valid only for the given number of days after it was defined. Once a password is expired, it can be used only exactly

one more time to login, because during this final login, the user will be forced to define a new password (which then again is valid for the given duration).

You can enable this option as the administrator in the Administration Hub (HUB): Go to **Main Menu → Security Options**.

Note: This setting applies only to LISTSERV Maestro accounts and administrators. It does not apply to subscriber accounts. Subscriber account passwords never expire.

Use With Care: The usefulness of forcing users to change their passwords on a regular schedule is questionable. In many cases, it simply induces users to create passwords that follow predictable patterns (e.g. incrementing a number, changing a letter to a similar symbol, adding or deleting special characters, switching the order of digits, etc.). This of course cancels the hoped for security gains of the forced password change because an attacker who gains the old password can often easily deduce the modified current password from it.

It is therefore recommended that you only enable the automatic password expiration option if you have good reasons for this.

1.4 Protection Against Dictionary Attacks

A dictionary attack is a technique to gain illegal access to a system by employing a list of words in a dictionary automatically to determine the login password for a given user account. The effectiveness of such an attack can be reduced by only allowing a limited number of invalid login attempts and by locking access to the account for a certain time. (Locking means that the login is denied even if the correct password is supplied.) LISTSERV Maestro supports this form of login locking for standard LISTSERV Maestro accounts and administrator accounts. Login locking is not available for list subscriber accounts.

In a fresh LISTSERV Maestro installation, the maximum number of login attempts is set to 5, with a 5 minute lock time if the maximum attempts are exceeded.

To access this option, Got to **Main Menu → Security Options**.

If a LISTSERV Maestro user or administrator account is locked due to too many login attempts and you cannot wait for the lockout duration to elapse, you can unlock all currently locked user and administrator accounts by clicking the "*Unlock all currently locked accounts*" button.

If it should happen that also all administrator accounts are locked due to too many login attempts, then of course you cannot login as an administrator to click this button. In this case, if you cannot wait for the lockout duration to elapse, you can unlock the administrator login by adding the following entry to the `hub.ini` file:

```
UnlockLockedAccess=true
```

With this entry in the `hub.ini`, login with an administrator account using the direct HUB login URL:

```
http://YOUR_SERVER/hub/?loginOverride
```

Provided that you supply the correct password for the administrator account, with the entry in the `hub.ini`, this next login will be accepted even though the account is currently locked out. Additionally, the administrator account in question is then no longer locked. Also, the entry from the `hub.ini` file is removed automatically during this login (or even if an unsuccessful login attempt is made).

As a final resort, if a system restart is an option, you can also restart the system to unlock all locked user and administrator accounts.

2 Disallow Concurrent Access with the Same User Account

The administrator has the option of allowing or disallowing users to log in twice with the same user account at the same time.

The default is, that such concurrent access is allowed.

From LISTSERV Maestro's point of view, there is no good reason to disallow concurrent access: Even if the same account is used several times for log in at the same time, each login session will be handled totally separately and the different sessions will not interfere with each other, just as if they were different sessions assigned to actually different accounts.

In fact, because of the caveats associated with disallowing concurrent access (see below), it is recommend to keep the default behavior and allowed concurrent logins, unless there are very good reasons against it (usually on an organizational level).

If you are considering to disallow concurrent access for reasons of accountability, i.e. because you want to be able to see the username in the log file and "know" which actual person was active with that account (which may not be possible if people share an account and even log in at the same time with that account), then instead of disallowing concurrent access (with the same account), you should probably instead decide, to give every person its own separate account with a secret password, which is not to be shared with other persons. That way, if you see a certain account mentioned in the log file, you "know" that it was the corresponding person which was active at the time. However, it does not matter if that person was logged in only once with the given account, or probably several times (in different browser windows) – it was still the same person.

Actually, the ability to be logged in several times with the same account at the same time might even be a useful thing, for example to be able to view one page of LISTSERV Maestro (like an overview page) and at the same time work actively in a different area of LISTSERV Maestro.

If for some reason you still want to disallow concurrent access, you can do so in the Administration Hub: Go to **Main Menu → LUI Settings → General Administration** and check the corresponding option.

If concurrent access is disallowed in this way, the Maestro User Interface will then behave as follows:

If a user logs in with a certain account, and another user is already logged in with the same account, the system will not accept the second login right away, but will instead do the following:

- If the second login attempt comes from a different workstation, the user doing the second login is simply informed that "Someone is already logged in with the same account from a different workstation, your login will not be accepted". The user is not logged in. However, the user may still use a different account (which is not currently in use) to log in.
- If the second login attempt comes from the same workstation, the user is informed, that already a previous session is active from the same workstation. The user is then queried, if he wants to cancel the second login, or if he wants to proceed with the second login and instead log out the previous session.

  If the user cancels the second login, the previous session will be unaffected, but the second login attempt will fail.

  If the user does not cancel the second login, then he will indeed be logged in, but the previous session will be logged out.

  Such a second login attempt from the same workstation may happen in situations similar to the following two:

1. A user has one browser window open, in which the first login session is active. He now opens a second window and tries to log in again with the same account. In this case, he will be notified that there still is a session open from his workstation and that proceeding with the second login will log out that first session. He will probably cancel the second login instead and continue using the first session.

2. A user has been using a first login session in a browser and has closed the browser without logging out properly. Since the system has no way of knowing, that the user has closed the browser, it will still keep the user's login session active. And since the browser is closed already, the user has no way of "going back" to that session to log out properly.

   This is usually not a problem, since the system will log out the session automatically after a certain timeout has passed (usually 90 minutes). However, if in the meantime the user opens a new browser window and tries to log in again with the same account, he will be notified that there is already a session logged in from his workstation, and that proceeding with the second login will automatically log out that first session. Since the first session is the one which the user has no access to any longer anyway, he will most probably agree to this and simply proceed with the second login.

The determination if a second login attempt comes from the same or from a different workstation is made by looking at the IP-address of the workstation used to make that attempt.

This approach has a some caveats that you should be aware of:

- If a group of users which are accessing the Maestro User Interface is using a local subnet with local addresses, and a router with NAT (Network Address Translation) or some other method of address mapping is used to connect to the internet, and the Maestro User Interface is on the "other" side of that router, then to the Maestro User Interface, all users will appear to be using the same workstation, since they will all be having the same IP-address, namely that of the router.

  In that case, the Maestro User Interface will handle all login attempts as if they were originating from the same workstation, which may result in the following confusing or even harmful situation: One user is logged in with an account from workstation A. Now another user tries to log in with the same account, only from workstation B. Both workstations will appear to the Maestro User Interface as one and the same, since both will be using the same IP-address externally. The result is, that the second user will be notified, that already another session is active from his workstation with the same account and he will have the option of proceeding with the login and canceling the "previous" login. Only this other session would in fact be the session of the first user, i.e. by logging in, the second user would log out the first user, disrupting his work.

  To work around this situation, make sure that all users are using different accounts, and that the passwords are kept secret, so that no other user can use a colleague's account to log in from a different computer and thus log out that colleague.

- If a user is connected to the internet with a dial-up modem connection as provided by most ISPs, where the workstation's IP-address is assigned dynamically each time the user connects, i.e. it will get a different IP-address each time, then the following situation may happen:

  The user opens a browser and logs into the Maestro User Interface with a certain account. He then closes the browser without logging out properly, i.e. the session will continue to be active until the timeout has expired. The user then disconnects his internet connection. Shortly thereafter, he remembers that he has to do something in the Maestro User Interface anyway, so he again connects to the internet, opens another browser and tries to log in with

the same account as before. Only this time, he very probably was assigned a different IP-address then the first time, so now, to the Maestro User Interface, he will look like he is using a different workstation. As a result, the Maestro User Interface will simply tell him, that the account is currently in use from a different workstation and will not accept a login with that account.

The user now has no other choice than waiting for the 90 minutes timeout to expire, before he can login again with the same account, because to cancel the previous login, he would have to access the Maestro User Interface using the same IP-address as before, which is more or less impossible with this kind of dynamic address assignment.

To work around this problem, the user should remember always to log out properly. Or if he has closed the browser accidentally without logging out, before he disconnects the modem, he should open a new browser, log in again (which cancels the previous session) and then log out properly.

In addition the administrator may configure the session timeout of the Maestro User Interface to be shorter than the default of 90 minutes, so that at the worst case, the user does at least not have to wait for that long.

The timeout for the Maestro User Interface is configured in the following file:

`[maestro_install_folder]/lui/WEB-INF/web.xml`

In this file, you will find an XML-entry for the session timeout:

```
<!-- 1.5 hrs session timeout -->
<session-config>
    <session-timeout>90</session-timeout>
</session-config>
```

The value of "`90`" determines the session timeout in minutes. Set it to a suitable value, save the file and restart the Maestro User Interface.

(The same setting can be changed for the Administration Hub by editing the file `[maestro_install_folder]/hub/WEB-INF/web.xml`".)

3 Restrict Access by Source IP-Address

For each component (e.g. the Administration Hub, Maestro User Interface and Maestro Tracker and also for the subscriber access pages of the subscriber datasets), it is configurable who is allowed to access this component and who is not.

For this access restriction, the IP-address of the calling client is used (e.g. the address of the computer where the browser/e-mail-client is running, that is used to access the component).

This means, that it is for example possible to define, that everyone (e.g. all addresses) are allowed to access the Maestro Tracker component, but that only certain addresses (a local subnet, for example) are allowed to access the Maestro User Interface and Administration Hub components.

If access is not allowed for a certain address, then a client from that address will get a "`403: Forbidden`" error when trying to access the restricted component.

By default, no component access restrictions are in effect. To add access restrictions, you need to add one or several "`Restrict.CONTEXT.N`" entries into the file:

`[maestro_install_folder]/conf/tomcat.ini`

Each such entry must look something like this:

```
Restrict.CONTEXT.ID=NETWORK/MASK
```

with the following replacements:

`CONTEXT` : Replace with the context name for which you want to introduce a restriction. Usually you will probably want to restrict access to the Maestro User Interface and/or the Administration Hub, for which the context names are "`lui`" and "`hub`" respectively.

LISTSERV Maestro also uses the contexts "`trk`", "`list`" and "`z`" (for tracked links, for the subscriber pages of subscriber lists, and for shortened URL's) but usually it does not make sense to restrict access to them, as they should always be publicly accessible.

If you are using LISTSERV Maestro's Tomcat to also serve LISTSERV's WA (see "LMA Admin Tech Doc 12 – Adding Content to the Tomcat Server"), then you can also restrict access to the two contexts used by the WA: "`archive`" and "`scripts`".

Finally, you might have added your own custom content in a separate context (see also "LMA Admin Tech Doc 12 – Adding Content to the Tomcat Server" for details). If you have added this custom content as a new context directly in the default "`webapps`" folder, then simply use the name of this context as the replacement for "`CONTEXT`". But if the context with the custom content was added inside of an individual "`webapps-MAIN_HOST_NAME`" folder, then use "`MAIN_HOST_NAME-CONTEXTNAME`" as the replacement instead (where you replace "`MAIN_HOST_NAME`" and "`CONTEXTNAME`" accordingly).

`ID` : Replace with any ID-string that uniquely identifies the "`Restrict`" entry from all other "`Restrict`" entries in the same context. Which kind of ID-string you use is up to you, but you should limit yourself to alpha-numeric characters and make sure that you do not use the same ID-string for two "`Restrict`" entries with the same context name (i.e. two "`Restrict`" entries must at least differ in their "`CONTEXT`" or in their "`ID`" value, there must never be two entries where both "`CONTEXT`" and "`ID`" are the same).

`NETWORK` : Replace with the dot-separated IP-address of the subnet to which you want to grant access to the given context (like "`192.168.1.0`").

`MASK` : Replace with the dot-separated subnet-mask for the subnet specified above (like "`255.255.255.0`").

It is important to understand that the listed IP-address ranges or addresses are the addresses which are **granted** access. All unlisted addresses are thus implicitly **denied** access to this context.

If no such restriction entry is present for a certain context at all, then access to this context is **unrestricted** (this is the default for all contexts after installation).

In other words: If for a context there is no entry at all, then access to that context is unrestricted. If there is at least one entry, then access to that context is restricted and access is allowed only for the address listed in the entry (or entries) of that context.

Important: Because of the way the Maestro Tracker functions (by accepting tracking events from mails sent all over the internet), the Maestro Tracker component must usually be accessible to everyone, i.e. you should not specify any restriction entry for the "`trk`" context.

For similar reasons, you should also not specify any restriction entry for the "`list`" context, so that everyone has access to the subscriber pages of the subscriber datasets (unless you have some good

reason to restrict access to these pages, for example if you are using them only for internal purposes).

After you have saved the modified `tomcat.ini`, you need to stop and restart LISTSERV Maestro to make it aware of the changes.

If you have distributed the components of LISTSERV Maestro to several servers, then you might need to edit the `tomcat.ini` file of several of these servers, depending on which components you want to restrict.

For example, if all three, the Administration Hub, the Maestro User Interface and Maestro Tracker are installed on separate servers, then you would typically not add a restriction entry on the Maestro Tracker server (since Maestro Tracker needs to be accessible to all), but you might want to add restriction entries both to the `tomcat.ini` of the Administration Hub (using the "`hub`" context) and of the Maestro User Interface server (using the "`lui`" context).

Note: Independent of any IP-address restrictions that are defined as explained in this section, access from the local IP-addresses, i.e. from the server where LISTSERV Maestro itself is installed, is **always** allowed.

Examples:

```
Restrict.lui.0=192.168.1.0/255.255.255.0
```

This would restrict access to the Maestro User Interface ("`lui`") and only allow access for computers in the subnet range `192.168.1.0` through `192.168.1.255`. Computers with any other IP address would not be allowed to access the Maestro User Interface.

Access to all other components (for example the Administration Hub, Maestro Tracker or the subscriber pages of subscriber lists) would remain unrestricted.

```
Restrict.lui.0=192.168.1.0/255.255.255.0
Restrict.lui.1=192.168.6.21/255.255.255.255
Restrict.hub.0=192.168.6.21/255.255.255.255
```

This would restrict access to the Maestro User Interface ("`lui`") and only allow access for computers in the same subnet range as above, and additionally also for the single computer with the address `192.168.6.21`.

Also, access to the Administration Hub ("`hub`") is restricted and access is allowed only for this one same computer with address `192.168.6.21`.

Access to Maestro Tracker and the subscriber pages of subscriber lists remains unrestricted.

3.1 Advanced Configuration

The configuration options described up until here assume that the Maestro components are accessed through only one host name or that, if your Maestro binds to several host names, the access restrictions shall be the same for all host names. In advanced setups this is not necessarily true, i.e. you may be running your system with several host names (e.g. because different groups shall work isolated from each other, as is frequently the case in a hosting environment) and you may want to configure access restrictions specifically for one of these host names and may want to run other host names without restrictions or with different restrictions. To accommodate such advanced setups, the following more complex rule notation can be used:

```
Restrict.lui.0=target:*,allowed:192.168.1.0/255.255.255.0
```

As you can see, the part left of the "=" is the same as before. And the value to the right of the "=" now lists the target address(es) to which access is to be restricted, and then the allowed subnet that a client must belong to in order to be allowed access. The "*" as the target address(es) means that this restriction applies to *all* IP-addresses that Maestro is bound to, i.e. to access Maestro on *any* of its addresses, you must belong to the subnet that is specified by the "allowed:" part. As you can see, this advanced (optional) syntax defines exactly the same rule as the standard syntax that is described above.

With this advanced syntax, instead of specifying "*" as the target, you can also specify a specific target IP address, for example like this:

```
Restrict.lui.0=target:1.2.3.4,allowed:192.168.1.0/255.255.255.0
```

This entry has the effect that anyone who tries to access Maestro on the address 1.2.3.4 (the target address) has to belong to the allowed subnet as specified. Access to Maestro via any other address is **not** restricted (if Maestro is bound to other addresses).

Of course you can combine several of these entries to define different restrictions for different Maestro addresses, and also define several different allowed subnets for the same target address, for example like so:

```
Restrict.lui.0=target:1.2.3.4,allowed:192.168.77.0/255.255.255.0
Restrict.lui.1=target:1.2.3.4,allowed:192.168.88.0/255.255.255.0
Restrict.lui.2=target:5.6.7.8,allowed:192.168.101.0/255.255.255.0
```

With these three entries, if you want to access Maestro on 1.2.3.4, you must belong to the subnet 192.168.77.0 *or* the subnet 192.168.6.88.0 (because there are two entries with the target 1.2.3.4). But if you want to access Maestro on 5.6.7.8, you must belong to the subnet 192.168.101.0. And access to Maestro on any other address will be unrestricted.

Of course you can also combine the restrictions for specific target addresses with a general restriction for all (remaining) target addresses, like for example so:

```
Restrict.lui.0=target:1.2.3.4,allowed:192.168.77.0/255.255.255.0
Restrict.lui.1=target:1.2.3.4,allowed:192.168.88.0/255.255.255.0
Restrict.lui.2=target:5.6.7.8,allowed:192.168.101.0/255.255.255.0
Restrict.lui.3=target:*,allowed:192.168.222.0/255.255.255.0
```

Like above, this defines that to access Maestro on `1.2.3.4`, you have to belong to the subnet `192.168.77.0` or `192.168.88.0`. And to access Maestro on `5.6.7.8`, you have to belong to the subnet `192.168.101.0`.

But now, if you want to access Maestro on any other address (except `1.2.3.4` and `5.6.7.8`), you have to belong to `192.168.222.0`. All other access is denied!

The above uses the advanced "*" syntax for the global restriction for all addresses. You could also use the standard syntax, i.e. the following defines exactly the same rules as the previous example:

```
Restrict.lui.0=target:1.2.3.4,allowed:192.168.77.0/255.255.255.0
Restrict.lui.1=target:1.2.3.4,allowed:192.168.88.0/255.255.255.0
Restrict.lui.2=target:5.6.7.8,allowed:192.168.101.0/255.255.255.0
Restrict.lui.3=192.168.222.0/255.255.255.0
```

The order of the entries is irrelevant. When deciding which rules to apply for a given client request, Maestro does the following:

- It checks the IP address that the request is directed to (the target address).
- If there is a specific rule (or a set of rules) for this target address, then these target-specific rules are applied. If the source address of the request matches one of the allowed subnets in any of these specific rules, the request is accepted, otherwise it is denied. Any rules for other target addresses, as well any rules for the "`*`" target, are ignored.
- If there is no specific rule at all for the given target address, then Maestro instead uses the "`*`" rule(s), if any.

To repeat this: As soon as at least one specific rule for a given target address is defined, then any requests that are directed to this target address will only be tested against this specific rule (or set of rules). I.e. if the request does not satisfy the "allowed" condition of this specific rule(s), then it will be denied! There will be no defaulting to the global "`*`" rule in that case. The global "`*`" rule is only used for requests that are directed at a target address for which there is *no* specific rules at all. So you can have the following scenarios, depending on which "`Restrict`" rules are defined in tomcat.ini:

- No rules at all: All requests are accepted.
- Only global "`*`" rules (using either the standard or the advanced syntax): All requests are tested against these rules. A request is accepted if one of the rules is fulfilled. Otherwise it is denied.
- Only specific rules with specific target addresses (for one or several different specific target addresses): All requests that are directed at any of these specific target addresses are tested against the rules that match their target address. Such a request is accepted if one of the applicable specific rules is fulfilled. Otherwise it is denied. All other requests that are directed to addresses for which there is no specific rule are accepted.
- A mix of specific rules with specific target addresses and global "`*`" rules (standard or advanced syntax): All requests that are directed at any of the specific target addresses are tested against the rules that match their target address. Such a request is accepted if one of the applicable specific rules is fulfilled. Otherwise it is denied (even if there would exist a global "`*`" rule that the request would fulfill). All other requests that are directed to addresses for which there is no specific rule are tested again the global "`*`" rules. Such a request is accepted if one of the global rules if fulfilled. Otherwise it is denied.

4 Ignore Tracking Events From Proxies or Email Gateways

In recent years there has been a rise in the use of proxies and email gateways that not only scan incoming email for potential viruses in attachments but also follow links in the message to see if the sites that they lead to contain malware or that pre-load the images in the message. By itself, this functionality is a good thing since it yields an additional level of user protection. In the case of email marketing messages sent with LISTSERV Maestro, this however has the unwanted effect of generating tracking events that do **not** originate from the actual email recipient.

If you are a victim of such unwanted tracking events, then you have the option to ignore tracking events that come from a proxy or email gateway that is known for this behavior, if you know the IP-address of that proxy.

You can configure the Tracker component in a way much similar to how you would restrict access by IP-addresses (see previous section). With such a configuration, Tracker still accepts all requests, i.e. serves the open-up counting image and redirects to the configured target URL, to allow the proxy to scan the target page for malware content. However, if the request originated from one of the configured IP addresses, then Tracker does not count the event.

To configure Tracker for this behavior, you add entries to the tracker.ini file, similar to this:

```
DoNotTrack.0=192.168.6.21/255.255.255.255
DoNotTrack.1=192.168.7.17/255.255.255.255
```

The format for these entries (and the associated parsing logic) is almost exactly the same as described in the section above, the only difference is that you configure this in the file **tracker.ini** and use the "**DoNotTrack**" prefix (instead of "Restrict") without the additional context discriminator.

The advanced format described above is also supported, which means that you can configure different sets of email gateway addresses and/or subnets for different host names that your Maestro instance may be binding to. In the advanced format, however, you use the "matched:" token instead of the "allowed:" token, for example:

```
DoNotTrack.0=target:1.2.3.4,matched:192.168.77.17/255.255.255.255
```

5 Allowing Frame Embedding of LISTSERV Maestro Pages

Normally, LISTSERV Maestro includes the X-Frame-Options:SAMEORIGIN HTTP header with all responses. This tells the end user's browser, that it shall not allow any of the LISTSERV Maestro pages to be embedded, for example into an <iframe>.

This is a best practice behavior for security reasons, mainly to block clickjacking attacks.

For selective pages of LISTSERV Maestro, it may however be desired to actually allow frame embedding. The pages for which this is possible, and how, is explained in the following.

5.1 Allowing Frame Embedding of Subscriber Pages

In some situations it may be desirable to embed the pages of the public subscriber website of a subscriber list (or list group) into a parent page. For example, if this is the only possible method to integrate the subscriber pages with an existing website.

LISTSERV Maestro does not allow you to completely disable the embedding-protection of the subscriber pages. You can only selectively disable it for certain allowed frame ancestors. I.e., you need to specifically define which ancestors are allowed to embed the subscriber pages.

By default, no allowed ancestors are in effect. To add allowed ancestors, you need to add the "SubscriberPagesAllowedFrameAncestors" entry into the file:

```
[maestro_install_folder]/conf/tomcat.ini
```

The value of this entry must be a space separated list of allowed ancestor URLs, using the syntax and notation for the Content-Security-Policy:frame-ancestors HTTP header. Note, that the list that you provide here is directly converted into this header. This conversion also always includes the 'self' ancestor in the list, so you should *not* provide it yourself.

This means that if you provide this setting:

```
SubscriberPagesAllowedFrameAncestors=URL1 URL2 URL3
```

Then it will be converted into this HTTP header:

```
Content-Security-Policy: frame-ancestors 'self' URL1 URL2 URL3;
```

Which is then included with all subscriber pages.

Examples:

```
SubscriberPagesAllowedFrameAncestors=https://host.sample.com

SubscriberPagesAllowedFrameAncestors=https://*.sample.com https://*.sample.de
```

Attention: Please be aware of the security implications of using this feature. Supply only frame ancestors where the server is under your own control.

Note: Allowing certain ancestors to embed the subscriber pages does not work in older browser versions or in Internet Explorer. These browsers that do not understand the `Content-Security-Policy` header will ignore it and only see the `X-Frame-Options` header and thus disallow the subscriber page embedding. For any reasonably recent browser it should however work as expected.

Note: This setting in the `tomcat.ini` only has an effect on the server where the LUI component is installed. If this setting is present, then it affects all public website pages of all subscriber lists and list groups, of all user accounts on this server.

5.2 Allowing Frame Embedding of View-in-Browser Pages

In some situations it may also be desirable to embed the view-in-browser pages (i.e. the browser versions of the message contents) into a parent page. For example, if you want to show these view-in-browser pages as part of an archive of your mailings where the archive pulls the message contents into an embedded frame.

LISTSERV Maestro allows you to either completely disable the embedding-protection, or you can selectively disable it for certain allowed frame ancestors. I.e., you need to specifically define which ancestors are allowed to embed the subscriber pages.

By default, no allowed ancestors are in effect. To add allowed ancestors, you need to add the "`ViewInBrowserAllowedFrameAncestors`" entry into the file:

`[maestro_install_folder]/conf/tomcat.ini`

The value of this entry must either be the keyword `all` or a space separated list of allowed ancestor URLs, using the syntax and notation for the `Content-Security-Policy:frame-ancestors` HTTP header. Note, that (except for the keyword `all`) the list that you provide here is directly converted into this header. This conversion also always includes the `'self'` ancestor in the list, so you should *not* provide it yourself.

This means that if you provide this setting:

```
ViewInBrowserAllowedFrameAncestors=URL1 URL2 URL3
```

Then it will be converted into this HTTP header:

```
Content-Security-Policy: frame-ancestors 'self' URL1 URL2 URL3;
```

Which is then included with all view-in-browser pages.

If instead you supply the keyword `all`:

```
ViewInBrowserAllowedFrameAncestors=all
```

Then both the `Content-Security-Policy` and the `X-Frame-Options` headers are omitted and not included at all. As a result, there is no embedding-protection at all, even in older browsers.

Examples:

```
ViewInBrowserAllowedFrameAncestors=all

ViewInBrowserAllowedFrameAncestors=https://host.sample.com

ViewInBrowserAllowedFrameAncestors=https://*.sample.com https://*.sample.de
```

Attention: Please be aware of the security implications of using this feature. Supply only frame ancestors where the server is under your own control. Be aware that supplying the keyword `all` opens your view-in-browser pages to potential clickjacking attacks. Only use it if you understand what this means and have determined, that for you the risk and impact of such attacks is negligible.

Note: Except when using the keyword `all`, allowing certain ancestors to embed the subscriber pages does not work in older browser versions or in Internet Explorer. These browsers that do not understand the `Content-Security-Policy` header will ignore it and only see the `X-Frame-Options` header and thus disallow the subscriber page embedding. For any reasonably recent browser it should however work as expected.
If you want the embedding to work even in older browsers, you need to make the decision if completely dropping the embedding-protection is acceptable to you. If this is the case, then you can specify the keyword `all` to completely disable the protection.

Note: This setting in the `tomcat.ini` only has an effect on the server where the LUI component is installed. If this setting is present, then it affects all view-in-browser pages of all user accounts on this server.