Documentation › User manual › Registering agents › Using the registration service

Using the registration service

It's possible to register agents automatically with authd. Choose the method that best meets your needs:

Method		**Description**
Simple method		The easiest method. There is no authentication or host verification.
Use a password to authorize agents		Allows agents to authenticate via a shared password. This method is easy but does not perform host validation.
Verify manager via SSL		The manager's certificate is signed by a CA that agents use to validate the server. This may include host checking.
Verify agents via SSL	Host validation	The same as above, but the manager verifies the agent's certificate and address. There should be one certificate per agent.
	No host validation	The manager validates the agent by CA but not the host address. This method allows the use of a shared agent certificate.

Simple method

Get an SSL certificate

The first step is to get an SSL key and certificate. This is required in order to make authd work.

1. If you have a valid SSL certificate with its key, copy them into the *etc* folder:

```
# (Manager)
cp <ssl_cert> /var/ossec/etc/sslmanager.cert
cp <ssl_key> /var/ossec/etc/sslmanager.key
```

2. Otherwise, you can create a self-signed certificate:

```
# (Manager)
openssl req -x509 -batch -nodes -days 365 -newkey rsa:2048 -keyout /var/ossec/etc/sslmanager.key -out
/var/ossec/etc/sslmanager.cert
```

Register the agent

1. Start the authd server:

```
# (Manager)
/var/ossec/bin/ossec-authd
```

2. Run the auth client on the agent. You must enter the authd server's IP address, like this:

```
/var/ossec/bin/agent-auth -m 192.168.1.2
```

Some hints

By default, authd adds agents with a dynamic IP (like using "any" on `manage_agents`). If you want to add agents with static IP addresses, use `-i` at server-side:

```
# (Manager)
/var/ossec/bin/ossec-authd -i
```

On the other hand, **duplicate IPs are not allowed**, so an agent won't be added if there is already another agent registered with the same IP. By using the `-f` option, authd can be told to **force a registration** if it finds an older agent with the same IP - the older agent's registration will be deleted:

```
# (Manager)
/var/ossec/bin/ossec-authd -i -f 0
```

The `0` means the minimum time, in seconds, since the last connection of the old agent (the one to be deleted). In this case, `0` means to delete the old agent's registration regardless of how recently it has checked in.

Secure methods

Launching the authd daemon with default options would allow any agent to register itself, and then connect to a manager. The following options provide some mechanisms to authorize connections:

Method		Description
Use a password to authorize agents		Allows agents to authenticate via a shared password. This method is easy but does not perform host validation.
Verify manager via SSL		The manager's certificate is signed by a CA that agents use to validate the server. It may include host checking.
Verify agents via SSL	Host validation	The same as above, but the manager verifies the agent's certificate and address. There should be one certificate per agent.
	No host validation	The manager validates the agent by CA but not the host address. This method allows the use of a shared agent certificate.

Note

These methods can be combined.

Use a password to authorize agents

Note

Reference ossec-authd

The manager can be protected from unauthorized registrations by using a password. We can choose one ourselves or let authd generate a random password.

1. To specify a password manually, just write it to the file `etc/authd.pass`. For example, if the key were "TopSecret":

```
# (Manager)
echo "TopSecret" > /var/ossec/etc/authd.pass
/var/ossec/bin/ossec-authd -P

  Accepting connections. Using password specified on file: /var/ossec/etc/authd.pass
```

2. If you don't specify a password, then authd will create a password itself and tell you what it is:

```
# (Manager)
/var/ossec/bin/ossec-authd -P

  Accepting connections. Random password chosen for agent authentication: abcd1234
```

On the agent side, the key can be put in a file of the same name or specified as a command-line argument.

1. Using the file `etc/authd.pass`:

```
# (Agent)
echo "abcd1234" > /var/ossec/etc/authd.pass
/var/ossec/bin/agent-auth -m 192.168.1.2
```

2. Entering the password at the command line:

```
# (Agent)
/var/ossec/bin/agent-auth -m 192.168.1.2 -P "abcd1234"
```

Use SSL to verify hosts

Create a Certificate of Authority

First we are going to create a certificate of authority (CA) that we will use to sign the certificates for the manager and agents. Hosts will receive a copy of this certificate in order to verify the remote certificate:

```
openssl req -x509 -new -nodes -newkey rsa:2048 -keyout rootCA.key -out rootCA.pem -batch
```

⚠ Warning

The file `rootCA.key` that we have just created is the **private key** of the certificate of authority. It is needed to sign other certificates and it is critical to keep it secure. Note that we will never copy this file to other hosts.

Verify manager via SSL

1. Issue and sign a certificate for the authd server, entering the hostname or the IP address that agents will use to connect to the server. For example, if the server's IP is 192.168.1.2:

```
openssl req -new -nodes -newkey rsa:2048 -keyout sslmanager.key -out sslmanager.csr -subj
'/C=US/CN=192.168.1.2'
openssl x509 -req -days 365 -in sslmanager.csr -CA rootCA.pem -CAkey rootCA.key -out sslmanager.cert -
CAcreateserial
```

2. Copy the newly created certificate and the key to the manager's `etc` folder and start `ossec-authd`:

```
# (Manager)
cp sslmanager.cert sslmanager.key /var/ossec/etc
ossec-authd
```

3. Copy the CA (but not the key) to the agent's `etc` folder and run `agent-auth`:

```
# (Agent)
cp rootCA.pem /var/ossec/etc
agent-auth -m 192.168.1.2 -v /var/ossec/etc/rootCA.pem
```

Verify agents via SSL

Verify agents via SSL (no host validation)

In this example, we are going to create a certificate for agents without specifying their hostname, so that the same certificate can be used by many agents. This verifies that agents have a certificate signed by our CA, no matter where they are connecting from.

1. Issue and sign a certificate for the agent. Note that we will not enter the *common name* field:

   ```
   openssl req -new -nodes -newkey rsa:2048 -keyout sslagent.key -out sslagent.csr -batch
   openssl x509 -req -days 365 -in sslagent.csr -CA rootCA.pem -CAkey rootCA.key -out sslagent.cert -
   CAcreateserial
   ```

2. Copy the CA (but not the key) to the manager's `etc` folder (if not already there) and start `ossec-authd`:

   ```
   # (Manager)
   cp rootCA.pem /var/ossec/etc
   ossec-authd -v /var/ossec/etc/rootCA.pem
   ```

3. Copy the newly created certificate and key to the agent's `etc` folder and run `agent-auth`. For example, if the server's IP is 192.168.1.2:

   ```
   # (Agent)
   cp sslagent.cert sslagent.key /var/ossec/etc
   agent-auth -m 192.168.1.2 -x /var/ossec/etc/sslagent.cert -k /var/ossec/etc/sslagent.key
   ```

Verify agents via SSL (host validation)

This is an alternative method to the last section. In this case, we will bind the agent's certificate to the agent IP address as seen by the manager.

1. Issue and sign a certificate for the agent. Then enter its hostname or IP address into the *common name* field. For example, if the agent's IP is 192.168.1.3:

   ```
   openssl req -new -nodes -newkey rsa:2048 -keyout sslagent.key -out sslagent.csr -subj
   '/C=US/CN=192.168.1.3'
   openssl x509 -req -days 365 -in sslagent.csr -CA rootCA.pem -CAkey rootCA.key -out sslagent.cert -
   CAcreateserial
   ```

2. Copy the CA (but not the key) to the manager's `etc` folder (if not already there) and start `ossec-authd`. Note that we use the `-s` option in order to enable agent host verification:

   ```
   # (Manager)
   cp rootCA.pem /var/ossec/etc
   ossec-authd -v /var/ossec/etc/rootCA.pem -s
   ```

3. Copy the newly created certificate and key to the agent's `etc` folder and run `agent-auth`. For example, if the server's IP is 192.168.1.2:

   ```
   # (Agent)
   cp sslagent.cert sslagent.key /var/ossec/etc
   agent-auth -m 192.168.1.2 -x /var/ossec/etc/sslagent.cert -k /var/ossec/etc/sslagent.key
   ```

Forcing insertion

If you try to add an agent with an IP already listed in an existing registration, `ossec-authd` will generate an error. You can use the argument *-f* to force the insertion.

Example

We previously installed and registered the Wazuh agent on *Server1* with IP 10.0.0.10 and ID 005. For some reason, we then had to completely re-install *Server1* and thus we now need to install and reregister the Wazuh agent on *Server1*. In this case, we can use the "*-f 0*" parameter which results in the previous agent (005) being removed (with a backup) and a new agent being successfully registered. The new agent will have a new ID.