

# Configuration for Networks with Proxy or Local Authentication

This page describes how to configure Kiuwan to work with:

- Networks that use an internet proxy.
- Single Sign On authentication.
- A local authentication system.

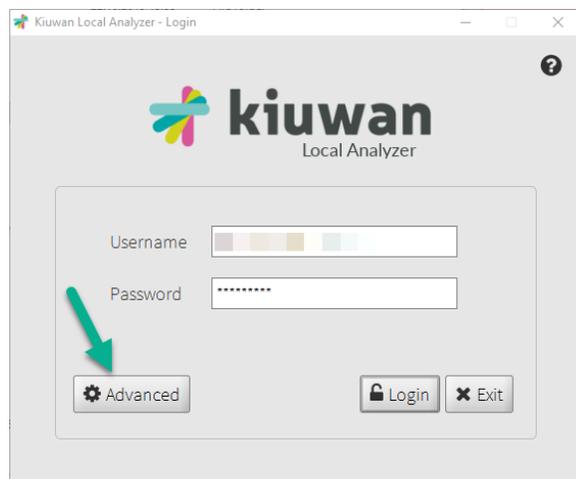
Contents:

- [How to configure Kiuwan for networks with proxy, SSO systems or local authentication](#)
  - [Internet proxy configuration](#)
  - [Single Sign-On](#)
    - [Available authentication methods for SSO](#)
      - [Windows Integrated Authentication \(WIA/IWA\)](#)
      - [HTTP authentication](#)
      - [Form-based authentication](#)
    - [Troubleshooting](#)
  - [Local authentication configuration](#)
    - [Configure with KLA GUI](#)
    - [Configure with CLI](#)
    - [Configure Kiuwan to use your Local Authentication system](#)
      - [A sample application](#)

## How to configure Kiuwan for networks with proxy, SSO systems or local authentication

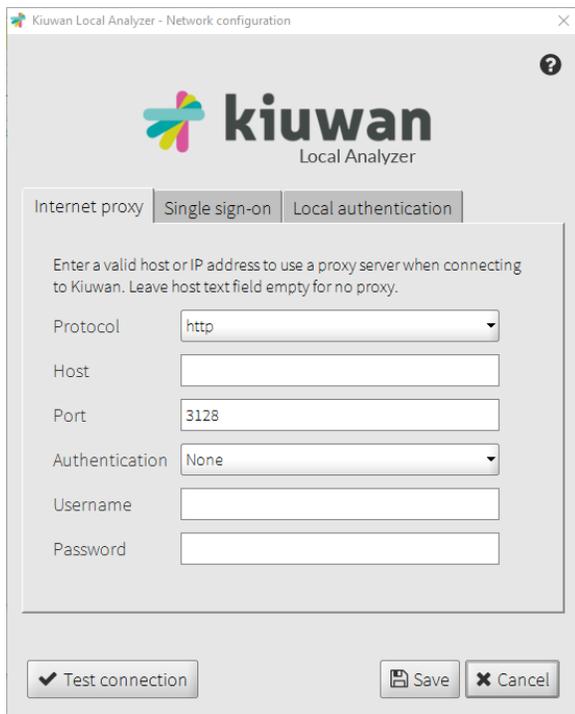
If your local network uses a proxy, or if your organization has a Single Sign On or a local authentication system, you can configure them from the log-in window:

1. [Start up the Kiuwan Local Analyzer](#)
2. Click **Advanced** to access the **Network configuration** window



### Internet proxy configuration

If your network has a pass-through proxy required for internet access, you should configure the proxy settings in the first tab of the **Network configuration window**.



Here you can set:

- The protocol used by the proxy (http or socks).
- The host of the proxy server (e.g. [proxy.myorganization.com](http://proxy.myorganization.com)). The port the proxy server listens to.
  - Leave this option empty if your connection to the Internet needs no proxy pass-through.
- The authentication type for the proxy.
- The username for the proxy authentication.
  - Only basic authentication is supported.
  - Leave this option to "None" if no authorization is required
- The password for the proxy authentication.

 From Java 8 Update 111 (8u111) onwards, the Basic authentication scheme has been deactivated, by default, in the Oracle Java Runtime. If you are using Java 8 Update 111 (or later), proxies requiring Basic authentication when setting up a tunnel for HTTPS will no longer succeed by default. Please visit [Basic Authentication Error : Proxy returns HTTP1.1 407 Proxy Authentication Required](#) for help on how to set up.

Alternatively, instead of using the Kiuwan Local Analyzer GUI, you can manually configure the proxy settings in `$(AGENT_HOME)/conf/agent.properties` file `agent.properties`

```
# HTTP(S) or SOCKS proxy configuration needed to access kiuwan. Leave
proxy.host empty if no proxy.
# One of 'http' or 'socks'.
proxy.protocol=http

# Proxy host. Leave empty if no proxy
proxy.host=

# Proxy port
proxy.port=3128

# Proxy timeout (milliseconds). Set to 0 for no timeout.
# Applies when establishing a connection or waiting for a response behind
a proxy.
proxy.timeout=

# Proxy authentication. One of 'None' or 'Basic'
proxy.authentication=

# Proxy username. Leave empty if authentication=none
proxy.username=

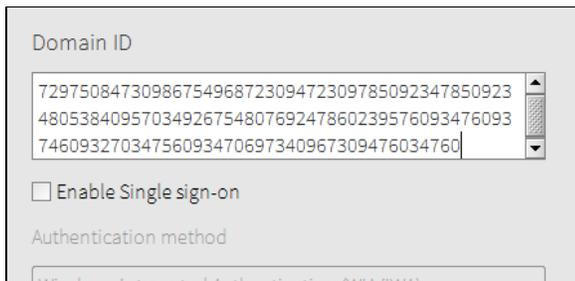
# Proxy password. Leave empty if authentication=none
proxy.password=
```

## Single Sign-On

Kiuwan Local Analyzer is able to authenticate the set user by accessing a Single Sign-On System. If your organization uses this kind of authentication, you can use this feature to use your SSO system credentials to access your Kiuwan account.

Please refer to [How to integrate Kiuwan with SAML SSO](#) for details on how SAML SSO works inside Kiuwan.

In order to configure Kiuwan Local Analyzer to use your SSO system, you need to indicate the Domain ID in the Single sign-on configuration panel to use any of the available authentication methods.



Domain ID

7297508473098675496872309472309785092347850923  
4805384095703492675480769247860239576093476093  
74609327034756093470697340967309476034760

Enable Single sign-on

Authentication method

## Available authentication methods for SSO

These are the available authentication methods for SSO.

### Windows Integrated Authentication (WIA/IWA)

If you are logged into the Domain account of your organization, KLA will authenticate against your ADFS via Kerberos using the credentials of your Windows session. You only need to be logged into the Windows Domain with your user.

Enable Single sign-on

Authentication method

Windows Integrated Authentication (WIA/IWA) ▼

IDP Hostname

\_\_\_\_\_

User-Agent to send during HTTP authentication

\_\_\_\_\_

Trusted hosts

\_\_\_\_\_

No Proxy Hosts

\_\_\_\_\_

These are the available parameters for this authentication method:

- IDP Hostname: The hostname of your organization's IDP. (e.g. [sts.contoso.com](http://sts.contoso.com))
- User-Agent to send during HTTP authentication: The User-Agent string to be sent to the IDP, you can leave this field empty if you want the KLA try to guess the suitable User-Agent. (e.g. MSIE 10.0)
- Trusted hosts: A comma-separated list of names, without spaces, of the hosts to allow the communication between the KLA and them. (e.g. [tok.organization.com](http://tok.organization.com),[sts2.myserver.com](http://sts2.myserver.com)) The IDP hostname (e.g. [sts.contoso.com](http://sts.contoso.com)) and SP (e.g. [www.kiuwan.com](http://www.kiuwan.com)) hostname are included in this list implicitly, so you can leave this field empty if you do not need to communicate with other servers during the SSO negotiation.
- No Proxy Hosts: A comma-separated list of names, without spaces, of the hosts to avoid to pass through the proxy server. (e.g. [sts.contoso.com](http://sts.contoso.com),[sts2.contoso.com](http://sts2.contoso.com)).

## HTTP authentication

If you cannot use WIA/IWA, but if your IDP supports HTTP authentication, you can choose this option.

Enable Single sign-on

Authentication method

HTTP Authentication ▼

IDP Hostname

\_\_\_\_\_

User-Agent to send during HTTP authentication

\_\_\_\_\_

Username to present to the IDP

\_\_\_\_\_

Password to present to the IDP

\_\_\_\_\_

Trusted hosts

\_\_\_\_\_

No Proxy Hosts

\_\_\_\_\_

These are the available configuration options:

- IDP Hostname: The hostname of your organization's IDP. (e.g. [sts.contoso.com](#)).
- User-Agent to send during HTTP authentication: The User-Agent string to be sent to the IDP, you can leave this field empty if you want the KLA try to guess the suitable User-Agent. (e.g. MSIE 10.0).
- Username to present to the IDP: Your username to authenticate against the IDP. (e.g. [jsmith@contoso.com](#)).
- Password to present to the IDP: Your password to authenticate against the IDP.
- Trusted hosts: A comma-separated list of names, without spaces, of the hosts to allow the communication between the KLA and them. (e.g. [tok.organization.com,sts2.myserver.com](#)) The IDP hostname (e.g. [sts.contoso.com](#)) and SP (e.g. [www.kiuwan.com](#)) hostname are included in this list implicitly, so you can leave this field empty if you do not need to communicate with other servers during the SSO negotiation.
- No Proxy Hosts: A comma-separated list of names, without spaces, of the hosts to avoid to pass through the proxy server. (e.g. [sts.contoso.com,sts2.contoso.com](#)).

## Form-based authentication

Select this option if none of the former options can be used in your environment, and if your IDP presents an HTML Form asking you for simply the username and password.

Enable Single sign-on  
Authentication method  
  
IDP Hostname  
  
Username to present to the IDP  
  
Password to present to the IDP  
  
Input field name for Username  
  
Input field name for Password  
  
Trusted hosts  
  
No Proxy Hosts

These are the available configuration options:

- IDP Hostname: The hostname of your organization's IDP. (e.g. [sts.contoso.com](#)).
- Username to present to the IDP: Your username to authenticate against the IDP. (e.g. [jsmith@contoso.com](#)).
- Password to present to the IDP: Your password to authenticate against the IDP.
- Input field name for Username: Name of the input field to fill in the HTML Form with your IDP's username. (e.g. the name of the input field for the username in ADFS 4.0 is 'UserName').
- Input field name for Password: Name of the input field to fill in the HTML Form with your IDP's password. (e.g. the name of the input field for the password in ADFS 4.0 is 'Password').
- Trusted hosts: A comma-separated list of names, without spaces, of the hosts to allow the communication between the KLA and them. (e.g. [tok.organization.com,sts2.myserver.com](#)) The IDP hostname (e.g. [sts.contoso.com](#)) and SP (e.g. [www.kiuwan.com](#)) hostname are included in this list implicitly, so you can leave this field empty if you do not need to communicate with other servers during the SSO negotiation.

- No Proxy Hosts: A comma-separated list of names, without spaces, of the hosts to avoid to pass through the proxy server. (e.g. [sts.contoso.com](http://sts.contoso.com),[sts2.contoso.com](http://sts2.contoso.com)).

## Troubleshooting

For troubleshooting please see [Troubleshooting Knowledge Base](#).

## Local authentication configuration

You can integrate the Kiuwan Local Analyzer with a local authentication system.

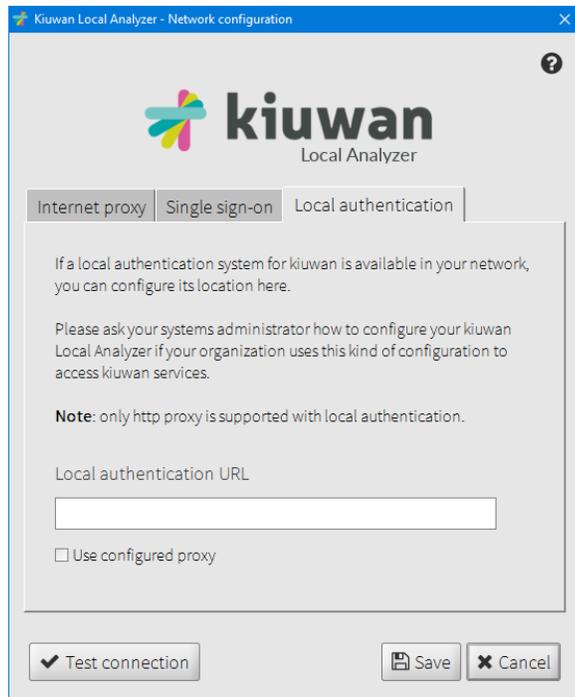
This is a common scenario in organizations that validate their employees' credentials against their own authentication system and do not want them to use other credentials when accessing external services.

If your company uses a corporate authentication service, your username and password will most probably be stored in an Active Directory, an OpenLDAP or an IBM Tivoli.

If that is your case, you do not need to have different credentials for your Kiuwan account.

By integrating Kiuwan with your Auth service, you will make the Kiuwan authentication delegate to your own system.

## Configure with KLA GUI



If this is the case, you will need to configure the URL of your organization's local authentication system. You can also set the local authentication to use the currently configured proxy (as long as it uses http protocol). Ask your systems administrator what value to enter in the "Local authentication URL" field.

When a local authentication URL is configured, the username and password you set in the login window will be sent to the local authentication system instead of Kiuwan.

## Configure with CLI

Alternatively, instead of using Kiuwan Local Analyzer GUI, you can manually configure the Local Authentication settings in `AGENT_HOME/conf/agent.properties` file `agent.properties`

```
# Integrate with a local authentication service
local.auth.url=
# Use the current proxy configuration when connecting to the local
authentication service (only valid when using 'http' proxy)
local.auth.useProxy=false
```

## Configure Kiuwan to use your Local Authentication system

Your company users should not connect to <https://www.kiuwan.com> to sign in, but to an **internal URL of your corporate network that you choose**, like: <http://kiuwan.yourdomain.com> or <http://yourdomain.com/kiuwan> (for example).

In that address you will have an authentication service application that will rely on your local auth service. If you have permissions to access Kiuwan, it will generate a **JWT authentication token** including the username, which is encrypted using a secret key, that you can generate in your Kiuwan account settings page.

This **token is sent to Kiuwan**, which makes the validation and creates the session for the user, who is **automatically redirected to <https://www.kiuwan.com>**, to access the application.

### A sample application

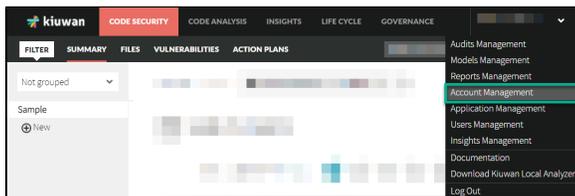
You can find a sample authentication application (kiuwan/kiuwan-local-authentication) as a way to get started.

This sample application uses Tomcat (tomcat-users.xml) as authentication mechanism. The steps are simple:

1. Install [Tomcat 8.5.11] (or another application server or use one you already have in your company) on a server.
2. Compile and deploy the sample authentication service application we provide for authenticating users in your application server.
3. Configure the authentication service application in index.jsp page. (Remember, this is a sample application. Do not use it as production code)

```
String ownerUsername = "puzzle307@gmail.com";
String clientId = "auth_1";
String secretKey =
"2chpil7khvun90irrrse2e2276v64s1pk8i9guh7ls544g3pjiiiv87763cfhgq62n6lvf7g
5liuvpteisr4lntnnh6q3dsik3j5j";
String kiuwanURL = "https://www.kiuwan.com/saas/web/dashboard/dashboard";
String loginURL = "http://localhost:8080/kiuwan-auth/login.jsp";
```

The required clientId and secretKey fields are generated from Kiuwan. You need login in Kiuwan and go to **Account Management > Secret keys**



You also need to configure the security settings in the application server where you deployed our authentication service application, to connect to your LDAP or any other authentication server.

In this example, we use Tomcat (tomcat-users.xml):tomcat-users.xml

```
<tomcat-users>
<role rolename="kiuwan_user"/> <!-- the role name as is named in web.xml
file of our authentication service -->
<user username="kuser" password="kuser" roles="kiuwan_user"/> <!-- kiuwan
users -->
</tomcat-users>
```

Configure the web.xml file to use this authentication mechanism:web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
<display-name>Kiuwan Authentication Service</display-name>
<!-- Define a Security Constraint on this Application -->
<security-constraint>
<web-resource-collection>
<web-resource-name>Kiuwan Authentication Service</web-resource-name>
<url-pattern>/index.jsp</url-pattern>
</web-resource-collection>
<auth-constraint>
<role-name>kiuwan_user</role-name>
</auth-constraint>
</security-constraint>
<login-config>
<auth-method>BASIC</auth-method>
</login-config>
<security-role>
<role-name>kiuwan_user</role-name>
</security-role>
<welcome-file-list>
<welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

Now you just have to tell your Kiuwan users to use the URL you have defined to access our authentication service application.

Remember that this same configuration is also valid if you have Single-Sign-On mechanisms such as LDAP, SPNEGO or IBM WebSeal.