SSO with Keycloak

Requirements

- FortiGate with at least FortiOS 7.X
- Keycloak (26.X ideally)

> This has been tested with Keycloak 26.X and FortiOS 7.6.6

> Realm names in Keycloak are case-sensitive. Ensure all URLs referencing the realm name use the exact same casing (e.g. `MyRealm` not `myrealm`).

> You need a separate Keycloak client for each FortiGate SSO use case. For example: one client for Admin SSO and one for User SSO (firewall policy authentication). This allows separate access restrictions per use case.

Keycloak Setup

The following steps apply to both the Admin SSO client and the User SSO client. Differences between the two are noted where applicable.

Step 1: Create the Keycloak Client (basic)

1. Log into Keycloak and go to *Clients*
2. Create a new Client of Type *SAML* and set the Client ID:
   - **Admin SSO:** `http://<your-fortigate-fqdn>/metadata/`
   - **User SSO:** `http://<your-fortigate-fqdn>/remote/saml/metadata/`

   > Note the trailing slash — it must be present. The Client ID must be a byte-for-byte match of the Issuer value sent by FortiGate in its SAMLRequest.
3. Finish the Create Client wizard with default settings for now

Step 2: Client Settings (advanced)

- General Settings
  - Set a display name for *Name*

    - Optionally set a *Logo URI* to display a logo on the account console. For FortiGate clients the official Fortinet logo is available at: `https://upload.wikimedia.org/wikipedia/commons/6/62/Fortinet_logo.svg`

      > The Logo URI is only shown in the Keycloak account console, not on the login page itself. To show a logo on the login page, a custom theme is required.

- Access settings
  - Set your FortiGate address as *Root URL* and *Home URL*
  - Set *Valid Redirect URIs*:
    - **Admin SSO:** `https://<your-fortigate-fqdn>/saml/?acs`
    - **User SSO:** `https://<your-fortigate-fqdn>/remote/saml/?acs`
  - Set *Master SAML Processing URL* to the same value as Valid Redirect URIs above
- SAML capabilities
  - Set *Name ID Format* to *username*
  - Enable *Force POST Binding*
- Signature and Encryption
  - Enable *Sign documents*
  - Enable *Sign assertions*
- Keys tab
  - Set *Client Signature Required* to *Off* — FortiGate does not sign AuthnRequests
- Logout settings (Settings tab)
  - Enable *Front channel logout*
  - Set *Logout service POST binding URL*:
    - **Admin SSO:** `https://<your-fortigate-fqdn>/saml/?sls`
    - **User SSO:** `https://<your-fortigate-fqdn>/remote/saml/?sls`
  - Set *Logout service Redirect binding URL* to the same value as above
- Logout settings (Advanced tab)
  - Set *Backchannel logout URL*:
    - **Admin SSO:** `https://<your-fortigate-fqdn>/saml/?sls`
    - **User SSO:** `https://<your-fortigate-fqdn>/remote/saml/?sls`
  - Enable *Backchannel logout session required*

Step 3: Add Username Attribute Mapper

Without a username mapper, FortiGate will reject the SAML assertion as it contains no username attribute.

1. Go to *Clients → your client → Client scopes → (client)-dedicated → Add mapper → By configuration*
2. Select *User Property* and configure:
   - *Name:* `username`
   - *Property:* `username`
   - *SAML Attribute Name:* `username`
   - *SAML Attribute NameFormat:* `Basic`
3. Save

Access Restriction (Optional)

By default, any user in Keycloak can authenticate through the client. To restrict access to a specific group, follow the steps below. Repeat this process for each client with its own role and group.

> This works by creating a realm role, assigning it to a group, and then creating a custom authentication flow that denies access to users who do not have that role.

Step 1: Create a Realm Role

1. Go to *Realm roles → Create role*
2. Set a name for the role (e.g. `Firewall Admins` or `Firewall Users`)
3. Save

Step 2: Assign the Role to a Group

1. Go to *Groups → your desired group → Role mapping*
2. Click *Assign role*
3. Select the role created in Step 1 and assign it

> All members of the group will automatically inherit the role. If you use AD/LDAP federation, make sure group sync is configured so group memberships are reflected in Keycloak.

Step 3: Create a Custom Authentication Flow

1. Go to *Authentication → Flows*
2. Click on *Browser → Action → Duplicate*
3. Give it a descriptive name (e.g. `Browser - Firewall Admins`) and confirm
4. Open the duplicated flow
5. Click *Add sub-flow* at the top level:
   - *Name:* `Role Check`
   - *Requirement:* `Conditional`
6. Inside the *Role Check* sub-flow, click *Add condition*:
   - Select *Condition - User Role*
   - *Requirement:* `Condition`
   - Click the settings (pencil) icon and set:
     - *Role:* select the role created in Step 1
     - *Negate:* `On`
7. Inside the *Role Check* sub-flow, click *Add step*:
   - Select *Deny Access*
   - *Requirement:* `Required`

> The Negate toggle is critical. With Negate On, the condition matches users who do**not** have the role, causing them to be denied. Without Negate, users **with** the role would be denied instead.

Step 4: Bind the Flow to the Client

1. Go to *Clients → your client → Advanced tab*
2. Scroll to *Authentication flow overrides*
3. Set *Browser flow* to your newly created flow
4. Save

FortiGate Setup

Preparation — Import Keycloak Certificate

> Make sure to download the certificate from your specific realm's metadata endpoint, not the master realm. Each realm has its own signing key.

1. Open `https://<your-keycloak-uri>/realms/<realm-name>/protocol/saml/descriptor` in your browser
2. Find the `<ds:X509Certificate>` value and copy it
3. Paste the content into a text file, wrapping it as follows:

```
-----BEGIN CERTIFICATE-----
<paste certificate content here>
-----END CERTIFICATE-----
```

4. Save the file with a `.crt` extension
5. Log into FortiGate and go to *System → Certificates*
6. Click *Create/Import → Remote Certificate*
7. Upload the file and note the assigned name (usually `REMOTE_Cert_X`)

Admin SSO Configuration

1. Log into FortiGate and go to *Security Fabric → Fabric Connectors*
2. Right-click on *Security Fabric Setup → Edit → Single Sign-On Settings*:
   - Set *Mode* to *Service Provider (SP)*
   - Click *Use current browser address* to automatically configure the SP Address
   - Configure the *Default admin profile* to your desired settings
   - IdP Settings:
     - Set *IdP type* to *Custom*
     - Set *IdP certificate* to the imported certificate name (e.g. `REMOTE_Cert_X`)
     - Set *IdP entity ID* to `https://<your-keycloak-uri>/realms/<realm-name>`

    - Set *IdP single sign-on URL* to `https://<your-keycloak-uri>/realms/<realm-name>/protocol/saml`
    - Set *IdP single logout URL* to `https://<your-keycloak-uri>/realms/<realm-name>/protocol/saml`

User SSO Configuration (Firewall Policy Authentication)

1. Log into FortiGate and go to *User & Authentication → Single Sign-On → Create New*
2. Configure the following:
    - Set *Type* to *Custom*
    - Set *Address* to your FortiGate FQDN or IP — the SP Entity ID, ACS URL, and SLO URL will auto-populate
    - Set *Attribute used to identify users* to `username`
    - Set *Attribute used to identify groups* to `group`
    - IdP Settings:
        - Set *IdP type* to *Custom*
        - Set *Certificate* to the imported certificate name (e.g. `REMOTE_Cert_X`)
        - Set *IdP entity ID* to `https://<your-keycloak-uri>/realms/<realm-name>`
        - Set *IdP single sign-on URL* to `https://<your-keycloak-uri>/realms/<realm-name>/protocol/saml`
        - Set *IdP single logout URL* to `https://<your-keycloak-uri>/realms/<realm-name>/protocol/saml`
3. Save the SSO configuration
4. Go to *User & Authentication → User Definition → Create New*:
    - Set *User Type* to *SAML*
    - Set *Username* to a name for this SAML user (e.g. `keycloak-users`)
    - Set *SAML SSO Server* to the SSO configuration created above
5. Go to *User & Authentication → User Groups → Create New*:
    - Add the SAML user definition as a *Member*
6. Apply the user group to a *Firewall Policy* under *Policy & Objects → Firewall Policy → Source*

> Users matching a firewall policy with this group will be prompted to authenticate via Keycloak SAML before traffic is allowed through.

Revision #3
Created 6 June 2026 15:26:56 by Oliver Karger
Updated 6 June 2026 16:13:02 by Oliver Karger