Evolveum

Introduction to Flexible Authentication

Lukáš Škublík / May 2024
Java Developer

Agenda

- Basic configuration
  - Modules
  - Sequence
- Examples
- The entire presentation was created for midPoint version 4.8.3

Flexible Authentication

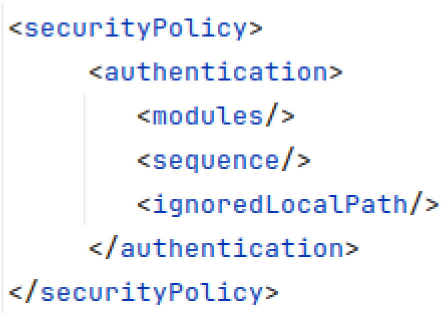
- Since midPoint 4.1
- Integration with other systems for authentication
- Used for more than just user login
- Different flow for different users

Flexible Authentication

- Many ways
- Using multiple authentication modules
- Combine modules to sequence
- Different sequences for each area

Security Policy

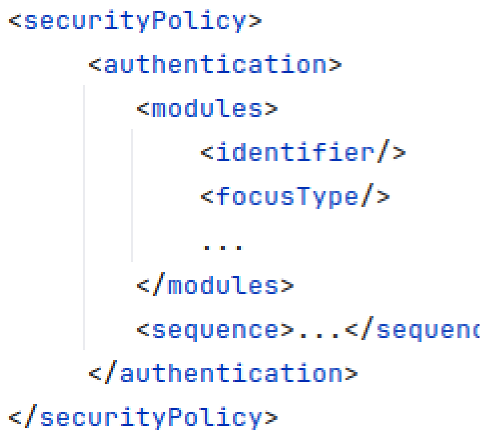
- Authentication
  - Modules
  - Sequence
  - IgnoredLocalPath
- System configuration type → globalSecurityPolicyRef



```
<securityPolicy>
    <authentication>
        <modules/>
        <sequence/>
        <ignoredLocalPath/>
    </authentication>
</securityPolicy>
```

Modules

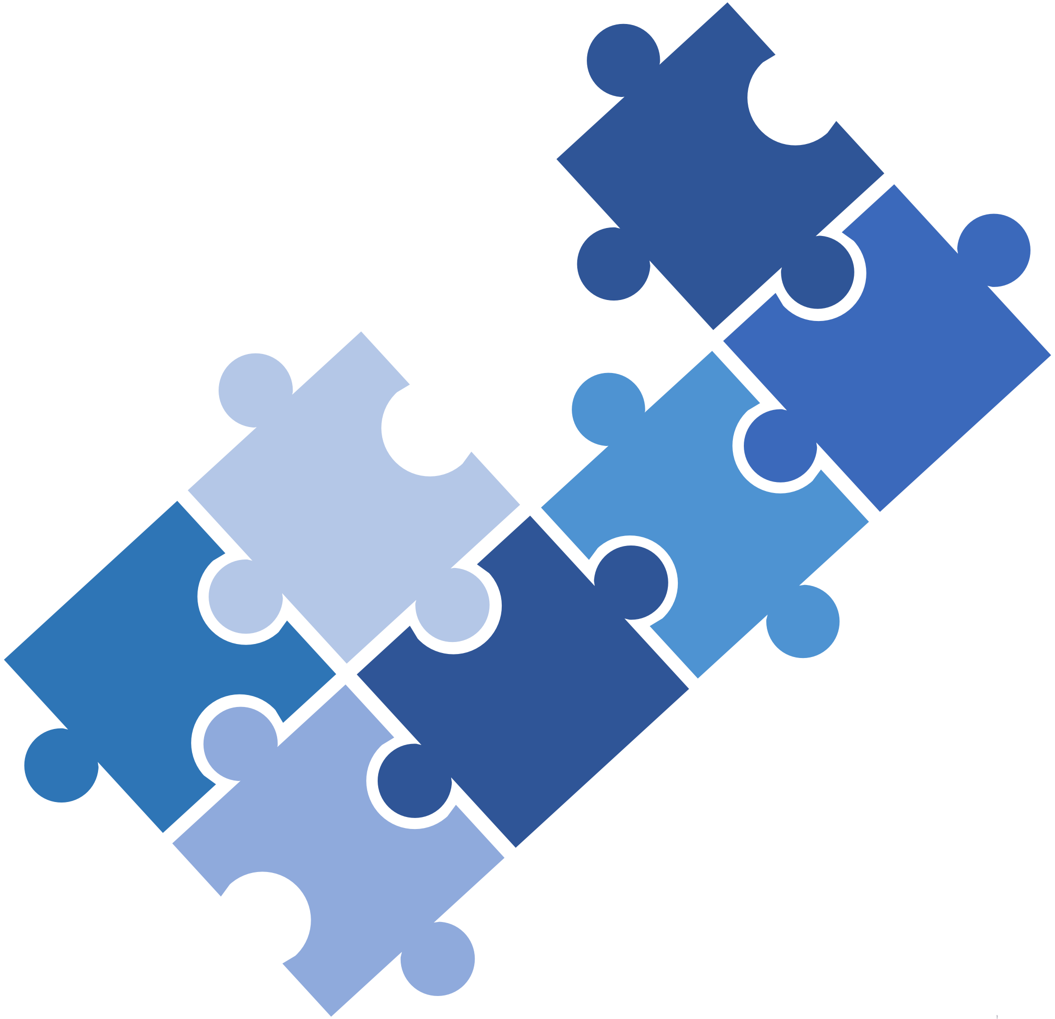
- Identifier
- FocusType
- Custom configuration for each module



```
<securityPolicy>
     <authentication>
        <modules>
            <identifier/>
            <focusType/>
            ...
        </modules>
        <sequence>...</sequenc
     </authentication>
</securityPolicy>
```

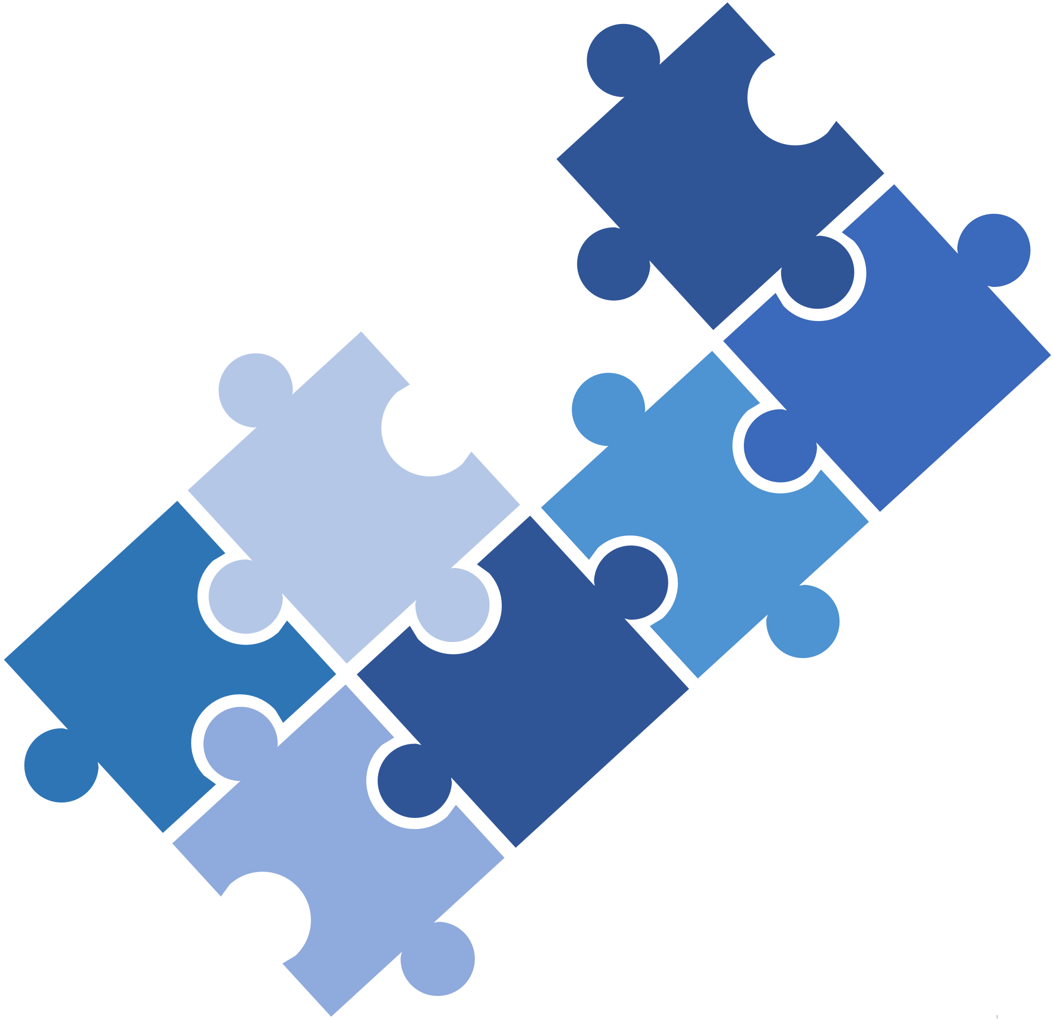
Modules

- Login form
- Http basic
- Http header
- Saml 2
- Oidc
- Mail nonce
- Ldap
- Duo



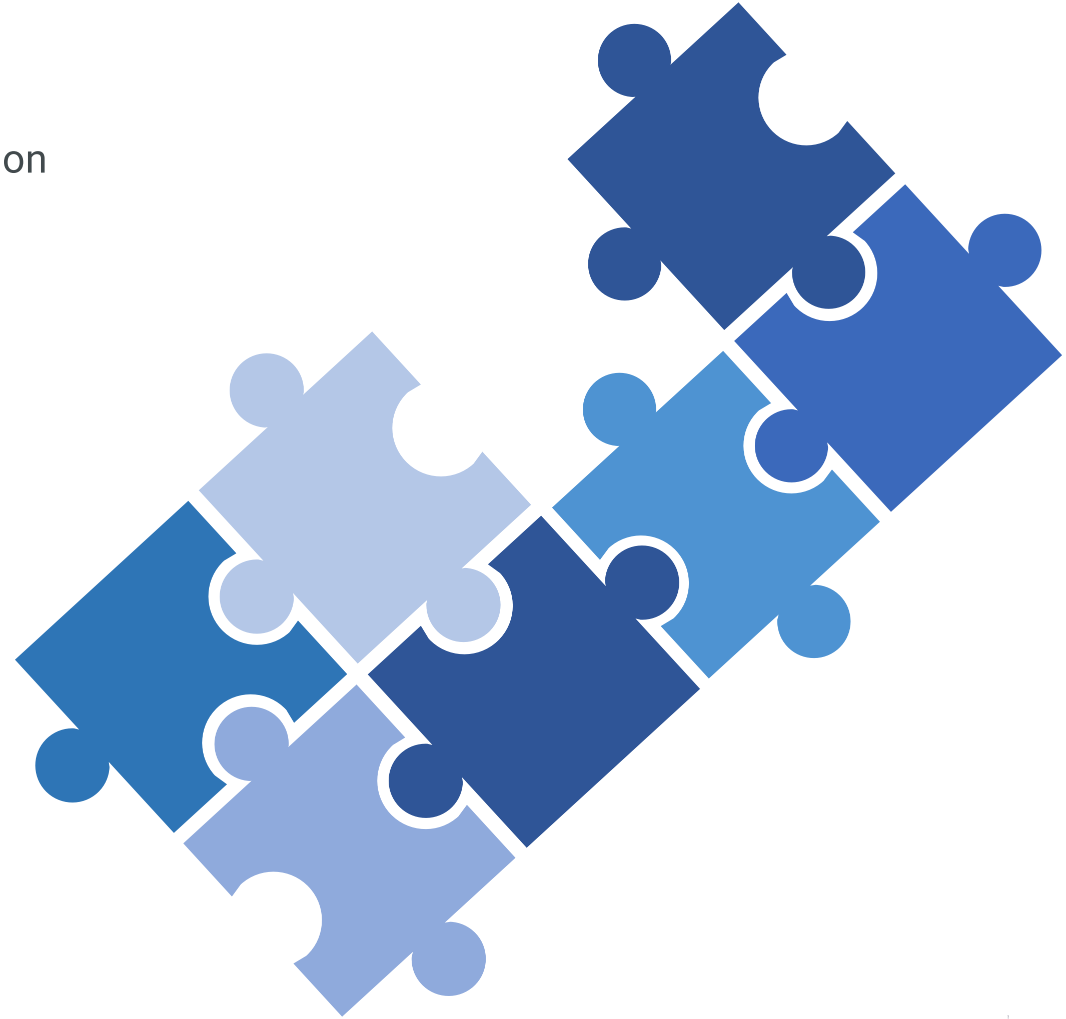
Modules

- Security questions form
- Http security questions
- Focus identification
- Archetype selection
- Hint
- Attribute verification
- Correlation



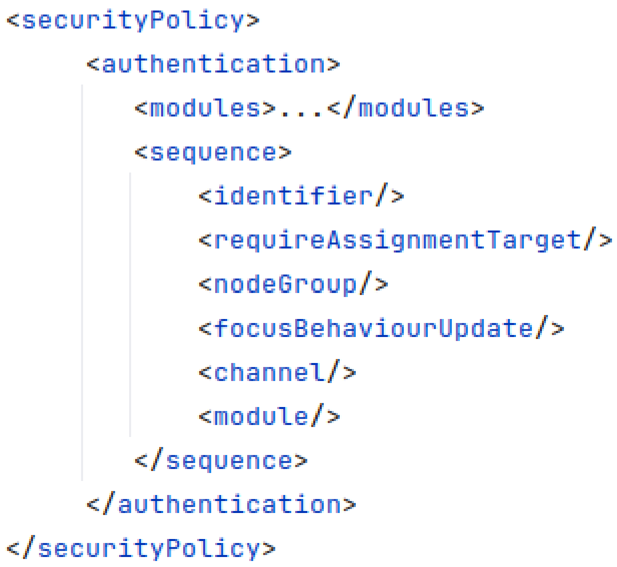
Modules

- Not considered sufficient for authentication
  - Focus identification
  - Archetype selection
  - Hint
  - Attribute verification
  - Duo
- Can’t be first in sequence
  - Duo
  - Attribute verification
  - Hint
  - Mail nonce



Sequence

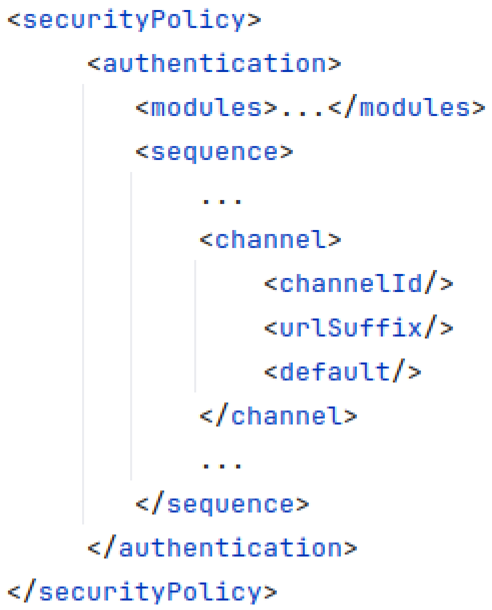
- Identifier
- RequireAssignmentTarget
- NodeGroup
- FocusBehaviourUpdate
- Channel
- Module



```
<securityPolicy>
     <authentication>
        <modules>...</modules>
        <sequence>
            <identifier/>
            <requireAssignmentTarget/>
            <nodeGroup/>
            <focusBehaviourUpdate/>
            <channel/>
            <module/>
        </sequence>
     </authentication>
</securityPolicy>
```

Sequence → Channel

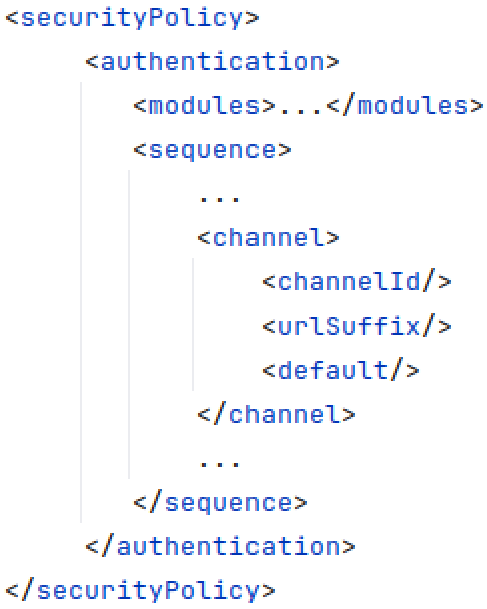
- ChannelId
  - http://midpoint.evolveum.com/xml/ns/public/common/channels-3#user
  - http://midpoint.evolveum.com/xml/ns/public/common/channels-3#rest
  - http://midpoint.evolveum.com/xml/ns/public/common/channels-3#actuator
  - http://midpoint.evolveum.com/xml/ns/public/common/channels-3#resetPassword
  - http://midpoint.evolveum.com/xml/ns/public/common/channels-3#selfRegistration
  - http://midpoint.evolveum.com/xml/ns/public/common/channels-3#invitation
  - http://midpoint.evolveum.com/xml/ns/public/common/channels-3#identityRecovery
- UrlSuffix
- Default



```
<securityPolicy>
    <authentication>
        <modules>...</modules>
        <sequence>
            ...
            <channel>
                <channelId/>
                <urlSuffix/>
                <default/>
            </channel>
            ...
        </sequence>
    </authentication>
</securityPolicy>
```

Sequence → Channel

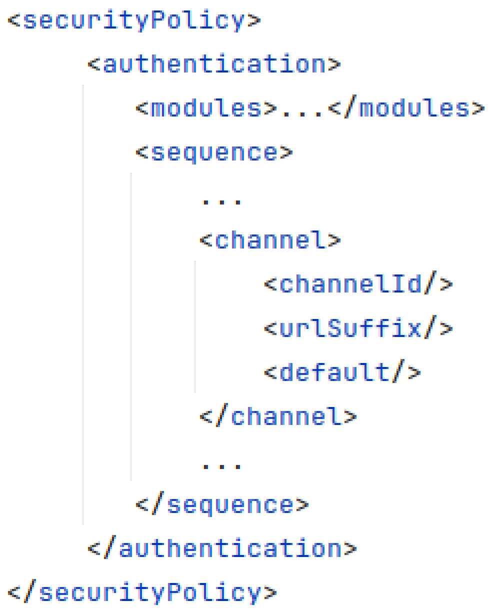
- ChannelId
- UrlSuffix
  - https://MIDPOINT_ADDRESS/midpoint/auth/URL_SUFFIX
  - https://MIDPOINT_ADDRESS/midpoint/auth/emergency
- Default



```
<securityPolicy>
     <authentication>
        <modules>...</modules>
        <sequence>
            ...
            <channel>
                <channelId/>
                <urlSuffix/>
                <default/>
            </channel>
            ...
        </sequence>
     </authentication>
</securityPolicy>
```

Sequence → Channel

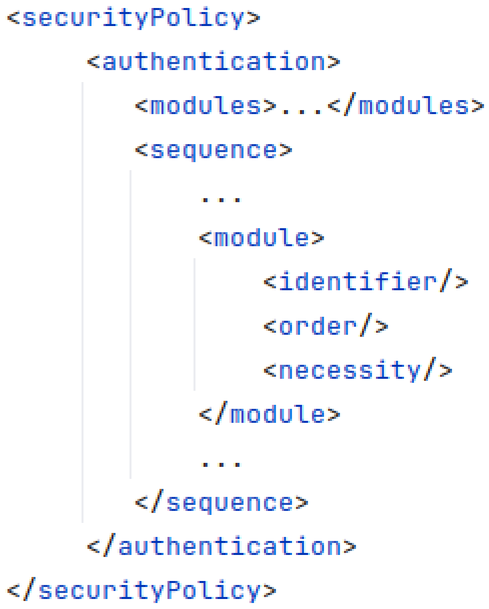
- ChannelId
- UrlSuffix
- Default



```
<securityPolicy>
     <authentication>
        <modules>...</modules>
        <sequence>
            ...
            <channel>
                <channelId/>
                <urlSuffix/>
                <default/>
            </channel>
            ...
        </sequence>
     </authentication>
</securityPolicy>
```

Sequence → Module

- Identifier
- Order
- Necessity
  - Sufficient
  - Required
  - Requisite
  - Optional



```
<securityPolicy>
    <authentication>
        <modules>...</modules>
        <sequence>
            ...
            <module>
                <identifier/>
                <order/>
                <necessity/>
            </module>
            ...
        </sequence>
    </authentication>
</securityPolicy>
```

Sequence → Module → Necessity

- SUFFICIENT
  - Sufficient to succeed
  - NOT required
  - Success → evaluation stop → result is a pass
  - Fail -> evaluation continues
- REQUIRED
- REQUISITE
- OPTIONAL

Sequence → Module → Necessity

- SUFFICIENT
- REQUIRED
  - Required
  - Success → evaluation continues
  - Fail → evaluation continues → result is failure.
- REQUISITE
- OPTIONAL

Sequence → Module → Necessity

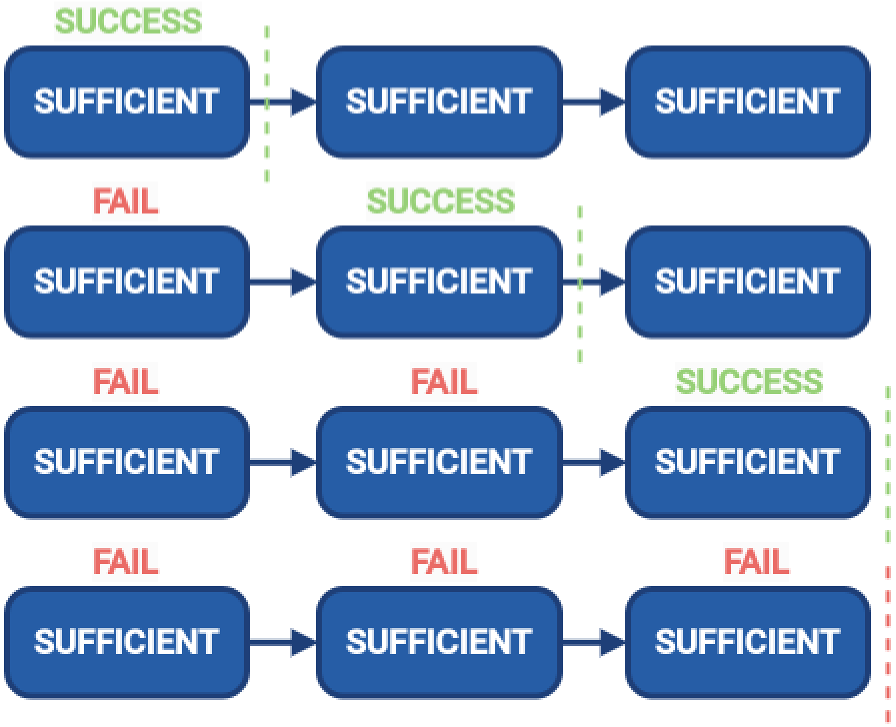
- SUFFICIENT
- REQUIRED
- REQUISITE
  - Required
  - Success → evaluation continues
  - Fail → evaluation stops with an error → result is failure
- OPTIONAL

Sequence → Module → Necessity

- SUFFICIENT
- REQUIRED
- REQUISITE
- OPTIONAL
  - Optional → NOT required → no real effect on the outcome
  - Success → evaluation continues
  - Fail → evaluation continues

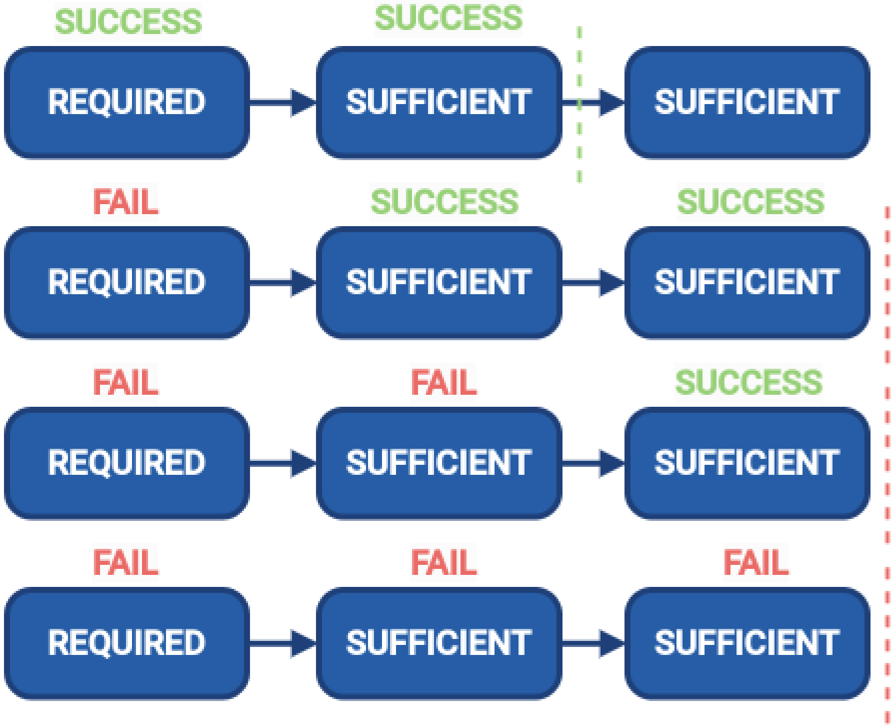
Sequence → Module → Necessity (Example)

- Any success → result is successful
- Any fail → evalution continue
- All fail → result is failed



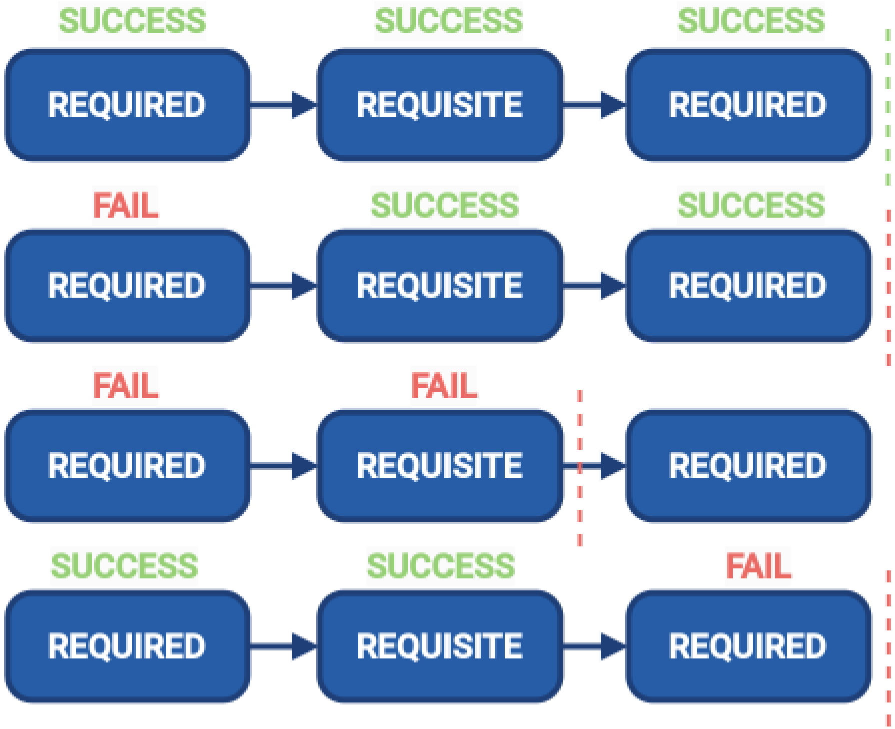
Sequence → Module → Necessity (Example)

- Required success or fail → evaluation continue
- Required and one of Sufficient success → result is successful
- Required fail → evaluation continue, but result is failed



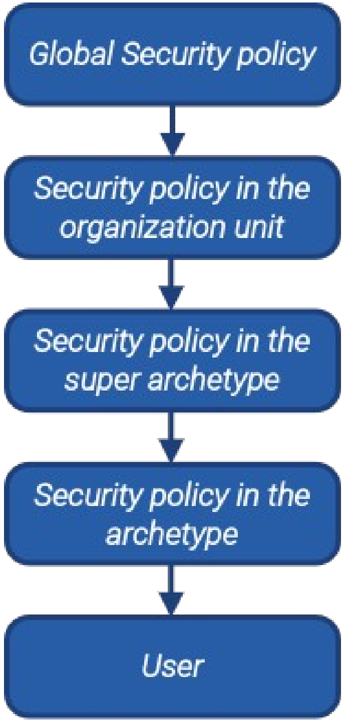
Sequence → Module → Necessity (Example)

- Required success or fail → evaluation continue
- Requisite fail → evaluation stop and result is failed
- Required fail → evaluation continu, but result is failed
- Result is successful only if all required and requisite modules success



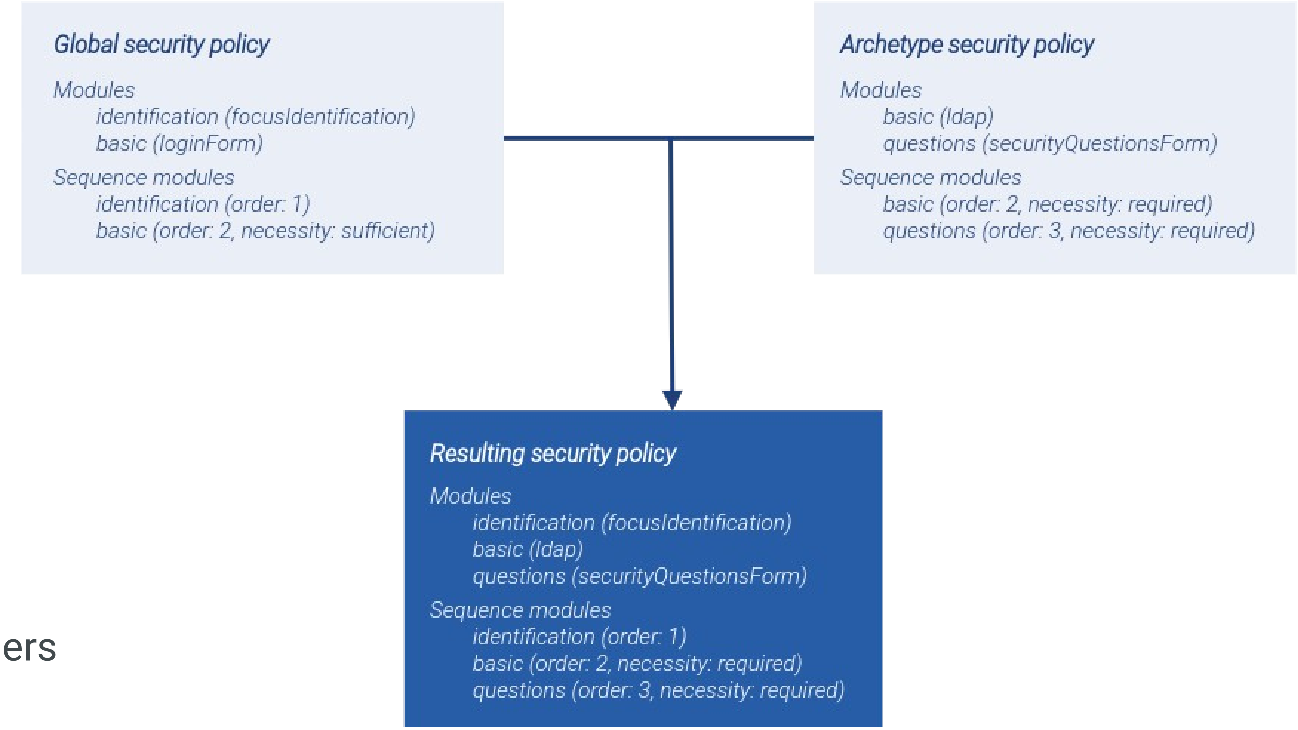
Merging Security Policies

- Since midPoint 4.7
- Security policies
  - Global in system configuration
  - In organization unit
  - In structural archetype
- Merging after identification
- Used sequence and module identifiers


Global Security policy
Security policy in the organization unit
Security policy in the super archetype
Security policy in the archetype
User

Merging Security Policies

- Since midPoint 4.7
- Security policies
  - Global in system configuration
  - In organization unit
  - In structural archetype
- Merging after identification
- Used sequence and module identifiers



Global security policy

Modules
- *identification (focusIdentification)*
- *basic (loginForm)*

Sequence modules
- *identification (order: 1)*
- *basic (order: 2, necessity: sufficient)*

Archetype security policy

Modules
- *basic (ldap)*
- *questions (securityQuestionsForm)*

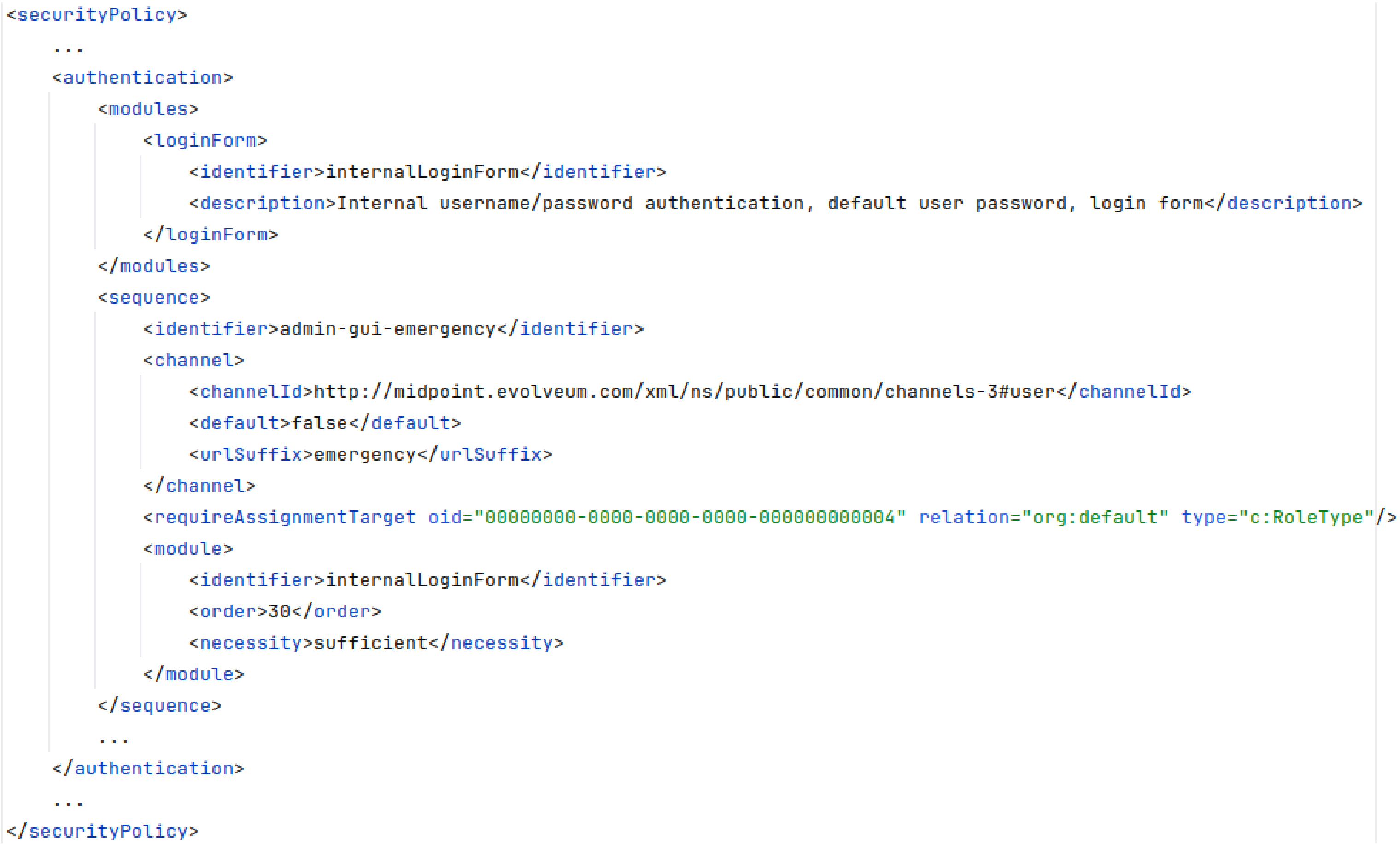
Sequence modules
- *basic (order: 2, necessity: required)*
- *questions (order: 3, necessity: required)*

Resulting security policy

Modules
- *identification (focusIdentification)*
- *basic (ldap)*
- *questions (securityQuestionsForm)*

Sequence modules
- *identification (order: 1)*
- *basic (order: 2, necessity: required)*
- *questions (order: 3, necessity: required)*

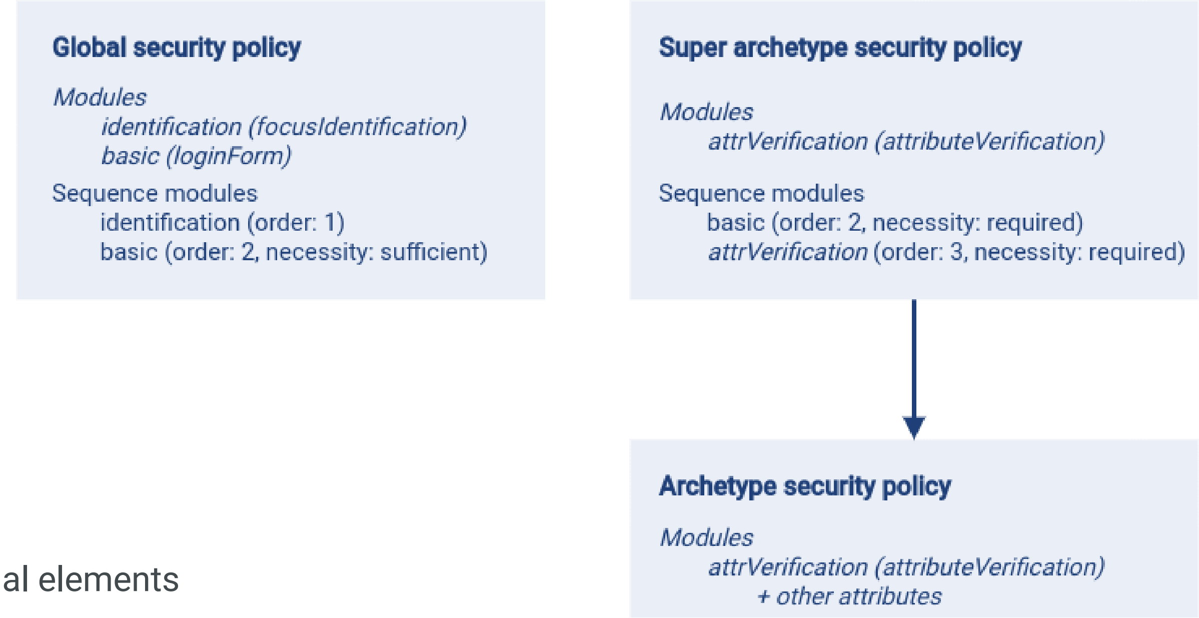
Simple Example



```xml
<securityPolicy>
    ...
    <authentication>
        <modules>
            <loginForm>
                <identifier>internalLoginForm</identifier>
                <description>Internal username/password authentication, default user password, login form</description>
            </loginForm>
        </modules>
        <sequence>
            <identifier>admin-gui-emergency</identifier>
            <channel>
                <channelId>http://midpoint.evolveum.com/xml/ns/public/common/channels-3#user</channelId>
                <default>false</default>
                <urlSuffix>emergency</urlSuffix>
            </channel>
            <requireAssignmentTarget oid="00000000-0000-0000-0000-000000000004" relation="org:default" type="c:RoleType"/>
            <module>
                <identifier>internalLoginForm</identifier>
                <order>30</order>
                <necessity>sufficient</necessity>
            </module>
        </sequence>
        ...
    </authentication>
    ...
</securityPolicy>
```

Complex Example

- Global security policy
  - Focus identification module
  - Login form module
- Super archetype
  - Attribute verification with one element
- Archetype
  - Expand Attribute verification with additional elements



Global security policy

Modules
identification (focusIdentification)
basic (loginForm)

Sequence modules
identification (order: 1)
basic (order: 2, necessity: sufficient)

Super archetype security policy

Modules
attrVerification (attributeVerification)

Sequence modules
basic (order: 2, necessity: required)
attrVerification (order: 3, necessity: required)

Archetype security policy

Modules
attrVerification (attributeVerification)
+ other attributes

Demo

Complex Example

Main Takeaways

- Flexible authentication provides more options for authentication.
- Flexible authentication can provide different authentication flows for different users.
- You can use it for more flows such as access to GUI, reset password, confirm registration, identity recovery and so on.
- Everything is in the docs!
  - https://docs.evolveum.com/midpoint/reference/support-4.8/security/authentication/flexible-authentication/configuration/

Next Webinars

- ISO27001 Compliance with MidPoint, May 30,2024
- Intermediate Configuration training teaser, June 20, 2024

Thank you for your attention

Do you have any **questions**? Feel free to contact us at **info@evolveum.com**

Follow us on social media or **join us** at GitHub or Gitter!

/Evolveum   @Evolveum   /Evolveum   /Evolveum   /Evolveum

Evolveum