

Evolveum

Securing MidPoint Deployments

Anton Tkáčik, April 2024
Software Developer

Agenda

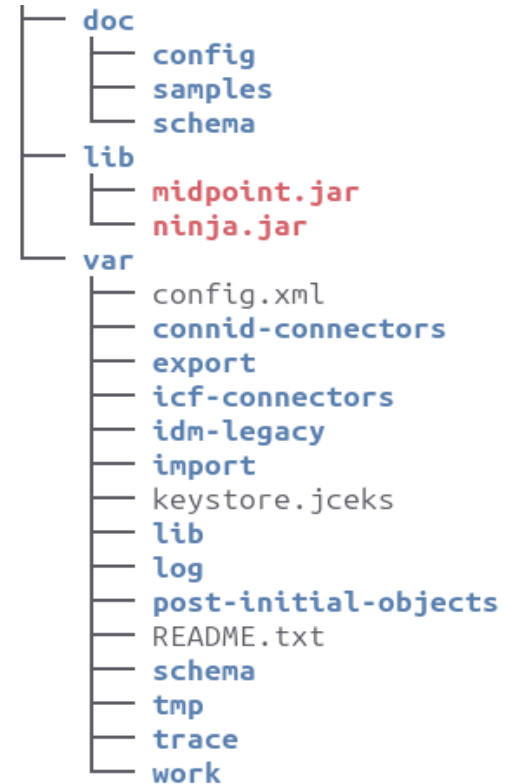
- Installation & Deployment
- Default Configuration
- Administrator and Superusers
- Mappings & Expressions
- Flexible Authentication
- Networking
- Penetration Testing



Installation & Deployment

Installation & Deployment

- Ensure that during the first run REST API and GUI are not exposed to public infrastructure.
- Ensure that only authorized accounts have filesystem access to midPoint home folder (usually **var** directory).
 - **lib**, **connid-connectors** and **icf-connectors** folders are used to load executable code.
 - **post-initial-objects** allows execution and import of midPoint objects and execution of bulk tasks with administrator privileges.
 - Database credentials, private encoding keys



Installation: Initial Administrator Password

- Since 4.8.1 initial administrator password does not have default value
- Initial administrator password is configurable, or it will be randomly generated
- You can customize it using **MP_SET_midpoint_administrator_initialPassword**
- Password must be 8 to 14 character long, contains at least one lowercase, uppercase and number. It may contain special characters.
- Even if you use configurable administrator password – change it after first login
- Learn more at [Docs: Administrator Initial Password](#)



Default Configuration

- Default configuration is mostly **intended to get you started**.
- Not the strictest security possible.
- **Review roles, security policy and password value policy** to see if they match your needs and roll out your own.
- You can find default configuration in doc/config/initial-objects inside midPoint distribution.
- Provided password policy is starting only – you should roll out your own one.



Administrator and Superuser Roles

- Change password for default **Administrator** user or **disable it**
- Do not use default **Administrator** user for day-to-day management of midPoint.
- You could use default Administrator as emergency account with very strong random password.
- We recommend you to use **Superuser** role for your own administrator accounts.
- Consider that **Superuser** role grants access to scripting – it is the same as granting shell access to midPoint VM / Docker.



Expressions & Scripting

Expressions & Scripting

- Mapping and expressions are one of the most powerful functions – that makes them also one of the **most dangerous** from security perspective.
- Avoid granting edit access to mappings and script expressions to non-technical non-administrator users.
- **Avoid granting access to Mapping Playground and Query Playground** to non-administrator users.
- If you need to grant edit access to expressions to non-administrator users, consider it very carefully and learn more about **Expression Profiles**.

```
<attribute id="5">
  <ref>ri:dn</ref>
  <displayName>Distinguished Name</displayName>
  <limitations id="44">
    <minOccurs>0</minOccurs>
  </limitations>
  <outbound>
    <source>
      <path>$focus/name</path>
    </source>
    <expression>
      <script>
        <code>'uid=' + name + ',ou=people</code>
      </script>
    </expression>
  </outbound>
</attribute>
```

Expression Profiles

- Expression profiles allows you to limit what particular expressions can do.
- You can limit expressions to use only subset of evaluators.
- You can disable access to Java / Groovy functions (currently compile-only, reflection access still possible).
- You can limit expressions to use only functions from specified function libraries.
- MidPoint bundles 1 expression profile – safe.
- See [Docs: Expression Profiles](#) for more about Expression Profiles.

```
<expressions>
  <expressionProfile id="218">
    <identifier>safe</identifier>
    <decision>deny</decision>
    <evaluator id="219">
      <type>asIs</type>
      <decision>allow</decision>
    </evaluator>
    <evaluator id="220">
      <type>path</type>
      <decision>allow</decision>
    </evaluator>
    <evaluator id="221">
      <type>value</type>
      <decision>allow</decision>
    </evaluator>
    <evaluator id="222">
      <type>const</type>
      <decision>allow</decision>
    </evaluator>
    <evaluator id="223">
      <type>script</type>
      <decision>deny</decision>
      <script id="224">
        <language>http://midpoint.evolveum.com/xml/ns/public
        <decision>allow</decision>
        <typeChecking>true</typeChecking>
        <permissionProfile>script-safe</permissionProfile>
      </script>
    </evaluator>
  </expressionProfile>
</permissionProfile id="225">
```

Expressions & Scripting – Code Injection

- MidPoint itself is protected from code injections via user entered data
- MidPoint does not sanitize user entered data – if you use HTML in description property – it will be there
 - MidPoint GUI will HTML-escape it
 - In mappings, expressions value will be unescaped
- You should know your target systems
- In your expressions avoid using any eval-like functionality with user entered data (groovy.util.Eval)

```
<expression>
  <script>
    <code>
import javax.naming.ldap.Rdn
import javax.naming.ldap.LdapName
import com.evolveum.midpoint.xml.ns._public.common.commonbase

dn = new LdapName('ou=Orgs,dc=example,dc=com')
parents = new ArrayList()
currentOrg = focus
while (currentOrg != null) {
  parents.add(currentOrg)
  // see com.evolveum.midpoint.model.impl.expr.Midpoint
  currentOrg = midpoint.getParentOrgByOrgType(currentOrg,
}

log.info("parents = {}", parents)

for (int i = parents.size() - 1 ; i >= 0; i--) {
  dn.add(new Rdn('ou', parents.get(i).name.toString()))
}

log.info("dn = {}", dn)

return dn.toString();
    </code>
  </script>
</expression>
```

Custom GUI Actions

- Before 4.8 using custom GUI actions required high privileges.
- Since 4.8 midPoint tracks origin of bulk actions and script expressions which allows them to be exposed and executed by less privileged users.
- In order to configure custom actions you need to configure: object collection, task templates, task template archetype to securely expose custom actions to non-privileged users.

For details see docs: [Trusted Bulk Actions](#)

```
<objectCollectionView>
  <identifier>allUsers</identifier>
  <type>UserType</type>
  <action>
    <name>reset-ad-password</name>
    <display>
      <label>Reset AD password</label>
    </display>
    <taskTemplateRef oid="a8fa6004-1ad7-445b-a54" />
  </action>
  <!-- ... -->
</objectCollectionView>
```

Flexible Authentication

Flexible Authentication

- Flexible Authentication is powerful feature, but misconfigurations may lead to security vulnerabilities
- Consider configuring emergency login if using remote authentication providers

```
<authentication>
  <modules>
    <loginForm id="1">
      <identifier>loginForm</identifier>
    </loginForm>
    <httpBasic id="2">
      <identifier>httpBasic</identifier>
    </httpBasic>
  </modules>
  <sequence id="3">
    <identifier>admin-gui-default</identifier>
    <displayName>Default gui sequence</displayName>
    <channel>
      <channelId>http://midpoint.evolveum.com/xml/ns
      <default>true</default>
      <urlSuffix>gui-default</urlSuffix>
    </channel>
    <module id="6">
      <identifier>loginForm</identifier>
      <order>1</order>
      <necessity>sufficient</necessity>
    </module>
  </sequence>
  <sequence id="4">
    <identifier>rest-default</identifier>
    <channel>
      <channelId>http://midpoint.evolveum.com/xml/ns
      <default>true</default>
      <urlSuffix>rest-default</urlSuffix>
    </channel>
  </sequence>
</authentication>
```

Identity Recovery, Password Reset

- **turned off by default**, but usually they are requested features for deployments.
- should be configured with care, since their presence allows for **Account Enumeration vulnerability**.
- Identities disclosed by Identity Recovery feature are audited for security reasons.
- Consider deploying rate-limiting proxy for Password Reset and/or Identity Recovery.



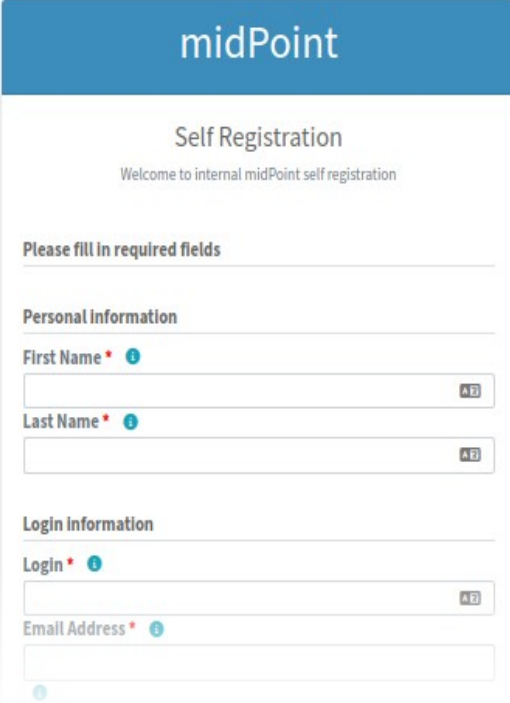
Security Questions

- **turned off by default**, but usually they are requested features for deployments.
- Try to use security questions combined with other type of authentication
 - Avoid using them as sufficient authentication
- avoid easily guesable searchable set of questions
 - organization structure, manager



Registration and Invitation Flow

- **turned off by default**, but usually they are requested features for deployments.
- Be careful – user creation or changes are done with elevated privileges
- Custom registration form
 - avoid exposing properties which have system-wide unique values (eg. login name)
 - avoid exposing properties which have specific lifecycle or policies attached
- Consider deploying rate-limiting proxy for registration.



The screenshot shows the 'midPoint Self Registration' interface. It features a blue header with the 'midPoint' logo. Below the header, the text 'Self Registration' and 'Welcome to internal midPoint self registration' is displayed. A section titled 'Please fill in required fields' contains three input fields: 'First Name *', 'Last Name *', and 'Login *'. Each field has a red asterisk and a blue information icon. Below these fields is an 'Email Address *' field, also with a red asterisk and a blue information icon. The form is styled with a clean, modern aesthetic and includes a small 'AD' icon in the bottom right corner of each input field.

Networking

HTTPS Proxy

- Always use **HTTPS Proxy** in production deployments.
- Make sure midPoint is **only accessible on port 443**, do not expose port 8080 (default midPoint HTTP port).
- More secure configuration
 - Configure **HTTP Strict Transport Security** if your proxy supports it
 - Optionally configure **Secure** attribute for **JSESSIONID** cookie

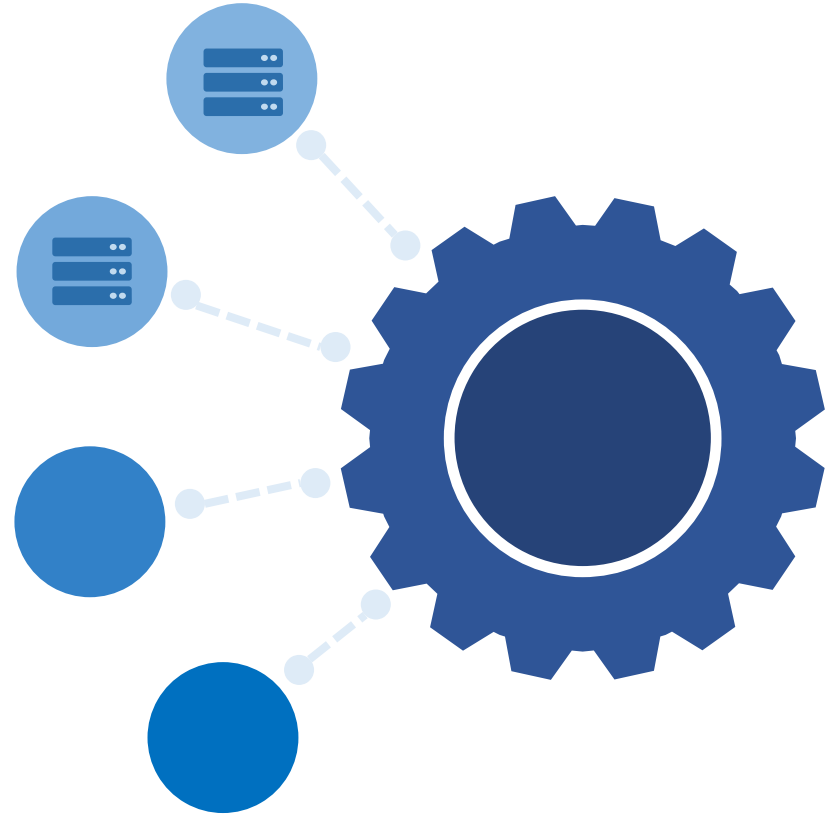


Rate-Limiting Proxy

- If you are using **Password Reset, Identity Recovery** or **Registration**, consider rate-limiting requests to these.
- MidPoint does not have built-in rate limiter, consider using 3rd party rate limiting proxy
- For Rate-Limiting key you can use **JSESSIONID** or IP address
- do not rate-limit whole midPoint, just endpoints for affected functionality.

Outbound Firewall

- **Outbound Firewall** - for very secure deployments.
- Do not forget that midPoint needs networking access to:
 - Database
 - Other midPoint nodes in cluster deployment
 - Resources



Penetration Testing, Security Fixes

Penetration Testing

- During development of MidPoint 4.8 was tested from security perspective
 - Testing done by Radically Open Security
 - Funded by NGI Zero Review grant
- Security-related fixes in 4.8, 4.8.1
- Various vulnerabilities were configuration-based
 - Consider doing penetration testing specific to your deployment



Security Fixes based on Penetration Testing

- Fixed in 4.8
 - Stored Cross-Site Scripting vulnerability (Advisory 019)
 - Click-jacking (embedding midPoint in iFrame)
- Fixed in 4.8.1
 - Stronger default password policy
 - Generated initial administrator password
 - Fixed CSRF logout (using bogus login to REST)
 - Fixed verbose error reporting for end users
 - Security vulnerability in invitation flow (Advisory 21)

Security Fixes

- Fixed in 4.8.2
 - Some users can execute script code beyond their authorizations (Advisory 22)
 - Some users can execute selected operations beyond their authorizations (Advisory 23)
 - Hidden panels on detail page are accessible by URL (Advisory 24)

Upcoming security features

- 4.9 - Secrets Providers
 - Allows loading secrets (resource passwords...) from external providers
 - Built-in providers: Docker provider, environment variables provider, file providers
 - Support for custom secrets providers
 - Search documentation for Secrets provider

Where to learn more?

- docs.evolveum.com
 - **MidPoint / Security**
 - **MidPoint / Security / Advisories**
 - **MidPoint/ Reference / Security**
- Midpoint Mailing List



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

© 2024 Evolveum s.r.o. All rights reserved.