Guidance for Service Provider Operators

- If an SP issues a request specifying use of the REFEDS MFA Profile, what can it expect from an IdP that also supports the Profile?
- If an SP issues a request specifying use of the REFEDS MFA Profile, what can it expect from an IdP that does not support the Profile?
- If an SP requests MFA using the REFEDS MFA Profile, does that alone guarantee MFA will happen?
- An SP prefers MFA, but does not want to require it. What should it do?
- Can the REFEDS MFA Profile be used with OpenID Connect?

If an SP issues a request specifying use of the REFEDS MFA Profile, what can it expect from an IdP that also supports the Profile?

An REFEDS MFA Profile-conforming IdP will reply to your login request in accordance with the SAML <RequestedAuthnContext> processing rules. If the IdP successfully authenticates the user in accordance with Profile requirements, it will respond with a SAML <Response> containing the https://refeds.org/profile/mfa in the <AuthnContextClassRef> element in its assertion:

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ...>
    ...
    <samlp:Status>
    </samlp:Status>
    <saml:Assertion>
        <saml:AuthnStatement AuthnInstant="...
            SessionNotOnOrAfter="...
            SessionIndex="...">
            <saml:AuthnContext>
                <saml:AuthnContextClassRef>https://refeds.org/profile/mfa</saml:AuthnContextClassRef>
            </saml:AuthnContext>
        </saml:AuthnStatement>
    </saml:Assertion>
</samlp:Response>
```

Example: Successful MFA

If instead the IdP is unable to perform MFA, and if it actually is able to respond at all, it will respond with a SAML <Response> with a second-level <StatusCode> of urn:oasis:names:tc:SAML:2.0:protocol:NoAuthnContext.

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ...>
    ...
    <samlp:Status>
        <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Responder"
    </samlp:Status>
</samlp:Response>
```

Example: Unsuccessful MFA

Note that a supporting IdP may not be able to support MFA for every user in every case. Your SP must anticipate this and handle errors accordingly: perhaps display a user friendly message informing them that MFA is required to access your service and was not possible. If the IdP provides an errorURL and/or support contact information in its SAML metadata, these may help the user navigate to the appropriate IdP support resources for additional assistance.
If an SP issues a request specifying use of the REFEDS MFA Profile, what can it expect from an IdP that does not support the Profile?

A SAML-compliant IdP MUST treat any unrecognized requested context class in the same way that it would treat one that was recognized but impossible to satisfy. Thus, the response, if issued, would contain the same error status noted above.

If an SP requests MFA using the REFEDS MFA Profile, does that alone guarantee MFA will happen?

No.

Most IdPs and campuses that support MFA services do not provide universal MFA coverage for their user communities. This means that even when a given IdP is capable of supporting the Profile, there is some probability that any given user may not be able to authenticate using MFA.

There is no defined mechanism at present to identify whether a given IdP is configured to support particular <AuthnContextClassRef> values, and SAML itself does not rely on that knowledge; it assumes that IdPs will respond in accordance with the standard when handling a request containing requirements it cannot meet. If the SP does not have any information about an IdP’s capabilities, it may not be able to distinguish between a case of specific users being unable to satisfy the profile, and an IdP as a whole not supporting it. Whether this distinction is relevant will depend on the SP.

The <AuthnContextClassRef> value is returned in the SAML assertion and that value is the only thing you can ultimately depend on and evaluate. It is not sufficient to configure an SP to request MFA and assume all non-error responses will therefore contain the MFA context. This is because users can generally bypass an SP’s SAML request configuration using unsolicited responses from an IdP, or by handcrafting a SAML request that does not include the MFA requirement.

It is therefore essential that an SP requiring MFA MUST enforce the presence of the value in the incoming assertion before proceeding; merely requesting the value does not ensure that the result will contain the proper value even if an assertion is returned, for a variety of reasons that are outside the scope of this FAQ. But you can assume that if such a thing were to happen, then either the IdP is not SAML-compliant, or that the request it saw was not the request you issued to it. Neither case is under the SP’s control.

How do I actually use the REFEDS MFA Profile to request MFA?

To request that a user be authenticated using MFA via this Profile, the SP must issue a SAML authentication request that includes the following <RequestedAuthnContext> content:

```xml
<RequestedAuthnContext>
  <AuthnContextClassRef>https://refeds.org/profile/mfa</AuthnContextClassRef>
</RequestedAuthnContext>
```

**IMPORTANT:** if your SP really requires MFA, include only one <RequestedAuthnContext> per the snippet above. Do not include other <AuthnContextClassRef> elements in your request.

While SAML permits an SP to request multiple authentication context classes, this only works reliably when the classes are truly distinct, and MFA often involves combinations of other pre-existing methods that can result in unusual behavior when one of those methods lines up with one of the values an SP requests.

In addition, the IdP can only respond with one requested authentication context; requesting more than one does not require that all of them are satisfied, only that one of them is. If you send more than one, the IdP has discretion and will choose (possibly in a way you would not expect) which requested context to satisfy. Order matters, but not in the way many people seem to expect it to.

Furthermore, even if an IdP supports the MFA Profile, it can only respond successfully to such a request if MFA is actually performed. If the user can authenticate to the IdP, but is not able to use MFA, the IdP MUST respond with an error. The SP will not receive any information about the user who tried to authenticate.

If this distinction is important, and it’s important to know the identity of the user even if MFA is not possible, consider the case described next (preferring MFA but accepting less). Application error messages when using this model should explicitly note that MFA is required to access the SP’s services. Note that most SPs do not provide sufficient default error handling capability for this and so this will generally require some effort.
An SP prefers MFA, but does not want to require it. What should it do?

Unfortunately, SAML 2.0 does not support this use case directly. It simply does not have the concept of "prefer but not require". While the list of context classes in a request is ordered (in fact this was clarified via errata), this is only useful when different types of authentication are not "composed" together, which is a very common situation with MFA deployments. MFA tends to combine password authentication with "other things" and so it is impossible to reliably ask for "MFA or Password" because one step happens first, satisfies the request, and may well short-circuit any further work in the interest of minimizing the user's burden.

Ordering can be useful when methods are discrete, such as Password vs. X.509 certificates vs. Kerberos /SPNEGO. In such a case, the IdP can be expected to try methods in order when it can, but even here IdPs may "prioritize" SSO so that a later method the user has already completed would be favored over an earlier method that would require a new challenge.

Nevertheless, some SPs prefer that a user authenticate with MFA but is willing to accept lesser approaches. Some scenarios where this might make sense:

- Applications that can implement a local scheme to do "stronger authentication" of specific users but prefer to allow users to use familiar campus mechanisms when available.
- Applications that will allow access to some services to all users, but have other services that are limited to those that authenticate using MFA.
- Applications that wish to offer their own opt-in feature for users to elect to use MFA for that service.
- An application that only allows access to users who authenticate with MFA, but wants to personalize error messages to users who do not use MFA as part of the authentication process.

Because this is not directly supported, there are a couple of alternative strategies to consider for these use cases, though neither is simple to implement.

"Step Up" MFA

The step-up model is often the most elegant if it can be orchestrated by the application and SP and is particularly well-suited to cases where either a minority of access needs MFA or where the use of MFA can naturally be required later in the interaction with the user.

With this model, the first interaction with the SP should be handled with a "vanilla" authentication request that does not stipulate anything regarding MFA and simply attempts to get the user identified and into the application.

Assuming this works, the application can proceed for some set of activities (or not), but eventually may either prompt the user to indicate whether MFA is possible, or just unilaterally attempt to elevate the authentication of the user by triggering a second authentication request back to the IdP with the requirement for MFA as described previously. Upon receipt of the new assertion, either the user's privileges can be elevated or in the event of an error, left as is with an indication to the user that the request for MFA was not successful.

Again, this may be very non-trivial with some application and SP software or integration patterns.

Note that while many IdPs and many MFA deployments are architected such that the second request for MFA may skip the password step and make for a relatively seamless step-up experience, this is by no means a guarantee. Some IdPs may not have the ability to do this, and some MFA technologies may rely on factors that don't include the same password.

MFA with Retry

Another common approach is to initially attempt authentication via MFA, but then handle an error from the IdP by downgrading and reissuing a request for regular authentication. In this scenario, the first request is issued with the additional content noted above, but if the IdP responds with an error, the SP has to detect this and either immediately reissue a request without the additional content, or present the user with an explanation and perhaps provide the option to do so.

This can be a bit more interoperable and result in fewer challenges to the user in some cases, but also requires more advanced error handling by the SP and application, so may not always be possible.

Can the REFEDS MFA Profile be used with OpenID Connect?

Eventually, almost certainly, but this is a preliminary answer because there is no official decision to directly adapt and express the profile URL into OIDC "acr" claims, the equivalent of SAML's `<AuthnContextClassRef>` element. It is likely this will happen, but consider that usage unofficial at this time.

In the event this happens, OIDC RPs are able to request specific "acr" values as "essential" claims, which is consistent with the various interactions described for SAML in other sections of this FAQ. Most of the same caveats also apply to the use of multiple values in requests and to IdP behavior in general.