Usage Guidance

For the InCommon “Base Level” and “MFA” Authentication Profiles

Overview

The InCommon Base Level Profile and InCommon MFA Profile define requirements that allow Service Providers (SPs) to request that Identity Providers (IdPs) perform multifactor authentication (MFA) as part of authenticating the current user. This Usage Guidance document provides non-normative information on the use of these profiles in practice. At the time of writing, the Profiles only define how they are to be used within SAML assertions and requests, but the intent is that the profile requirements should be extensible and relevant to other protocols, such as OpenID Connect.

The MFA authentication profile requires that MFA was used in a manner that mitigates certain risks to the authentication process compared to a single-factor authentication process. The MFA base level authentication profile does not identify the specific MFA technology used to mitigate those risks. This is to allow implementations to validate and rely on compliance with the profile overall, rather than be required to enumerate (and rank the strength of) every possible MFA technology as part of an authentication negotiation.

The (non-MFA) Base Level profile is defined solely to provide a value for systems to affirmatively assert when authentication is done successfully but *without* MFA (necessarily) being used, and is specific to supporting the details of InCommon SAML assertion and assertion request signaling.

While the profiles are written to be relevant for any multifactor authentication approach, much of the focus is on improving security of the authentication event when one of the factors is specifically the use of a username/password entered directly by the user.

General Guidance

Risks that must be mitigated

The MFA profile requires that “*single-factor-only risks related to non-real-time phishing, offline cracking, online guessing and theft of a (single) factor*” be mitigated. These terms are defined below.

Note that the MFA profile is generally addressing mitigation of credential theft that creates a *persistent threat to user authentication* for the targeted user. Theft that allows an attacker to masquerade as the victim for a very limited window of time or number of authentication events (e.g., phishing of a single one time password (OTP) value) are not required to be mitigated under this profile.

- Non-real-time phishing
    - Inducing a user to provide a credential (e.g., password) to a malicious agent through social engineering, forged websites or the like. While not technically "phishing", inducing a user to provide credentials using an insecure/observable protocol is also intended to be included as part of the risk to be mitigated. The profile uses the phrase *"non-real-time"* to emphasize that the protection is provided against indefinitely reusable credentials, and not one-time user attacks.

- Online guessing, offline cracking
    - Attacks that could obtain credentials without directly involving the user, through attacks on application login screens or (encrypted) password databases.

- Theft of a (single) factor
    - The ability to (steal) each factor using a single theft mechanism. The intent is that the factors should require different theft mechanisms. For example, stealing a user's phone or security FOB requires a different attack mechanism than phishing a user's password. (Again, note that theft of a single OTP code is not protected against, but rather theft of the OTP device).

What constitutes an acceptable "second factor"?

Specific Technologies

The MFA base level profile does not specify what specific technologies are sufficient to mitigate authentication risks. However, the authors have compiled a listing of some specific technologies and combinations of factors that are explicitly considered acceptable and not acceptable that can be used as a guideline to evaluate specific implementations. This listing can be found at MFA Technologies, Threats and Usage.

Types of Factors

The MFA profile specifies that the factors used to meet the MFA requirement must be of different "types", not just separate factors. This means that validating two separate passwords is NOT sufficient to meet the authentication requirements of this profile.

Independence of Factors

Implementors must work to ensure that the different factors used in the authentication process are independent, meaning that gaining access to one factor must not trivially grant access to the other factor.

- Any factor that is directly accessible using the first factor is no more secure than the single factor by itself, and so is NOT considered a second factor. Institutions are expected to provide safeguards to maintain the independence of their supported authentication factors
  - For example, a software/virtual phone that is authenticated using the enterprise password is not an appropriate second factor.
  - Additionally, users can take actions that reduce the ability to treat otherwise independent factors as "independent"; for example, a user storing their software OTP generator on a network device accessible using just the "first factor" password.
  - The MFA profile does not enumerate specific requirements the institution must meet to protect against these forms of authentication dependence, but technical restrictions (where feasible) and user education are highly recommended to mitigate the risks of users deploying factors in a manner that decreases their independence.

- Processes that allow a user to immediately register a new second factor (re-registration) using only their "first factor" enterprise password are no more secure than use of the enterprise password itself. Implementers are expected to require greater scrutiny before allowing registration of *replacement* or *additional* second factors to prevent attackers with password access from simply registering and immediately using a new second factor. However, the MFA profile does not provide any specific requirements on such registrations.
  - Note that it is common practice to allow the *initial* registration of a second factor using only the existing factor, and the MFA profile does not restrict this *initial* MFA factor registration practice.

SAML-Specific Guidance

The profiles are not specific to SAML, but this section provides guidance on their use in that context.

Representations in SAML

The recommended means of representing these profiles in a SAML assertion are via the `<AuthnContextClassRef>` element (SAML 2.0) or `AuthenticationMethod` attribute

(SAML 1.1). These are expressed in SAML statements used to represent acts of authentication by the subject of an assertion.

In the case of SAML 2.0, the use of the Authentication Context mechanism has the benefit of enabling signaling of requirements by a relying party in its requests to an identity provider, and the bulk of this section speaks to the use of this capability. The details given in the examples below all focus on usage under SAML 2.0.

Considerations when Requesting MFA AuthnContextClassRef Values

SP operators must understand that 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 this profile, there is a significant probability that any given user may not be able to authenticate using MFA.

A different IdP might be unable to fulfill any SAML requests involving these profiles because it is not configured to assert either `<AuthnContextClassRef>` value.

There is no defined mechanism at present to identify whether a given IdP is configured to assert either of these `<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.

SPs will need to validate that the `http://id.incommon.org/assurance/mfa` `<AuthnContextClassRef>` value is returned in SAML responses; it is not sufficient to configure an SP to request MFA and assume all 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 hand-crafting a SAML request that does not include the MFA requirement.

If an application intends to provide limited services to non-MFA authenticated users, the actual `<AuthnContextClassRef>` value returned to the SP will need to be evaluated dynamically by the application to determine the appropriate access to provide to the user.

Creating SAML Requests

From a technical standpoint, when generating a SAML authentication request where the MFA profile is desired, the approach is fairly straightforward:

1. Explicitly list every `AuthnContextClassRef` value that your SP is willing to accept in the <`RequestedAuthnContext>` element in your SAML request. The actual values you list will depend on your use case (see "Use Cases" below for some general guidance).

2. No matter how carefully you specify context class values, some IdPs may be unable to respond due to software or process limitations. *(This issue is not specific to the MFA profile but affects any requests that includes explicit <`RequestedAuthnContext>` elements.)*
    - If you want to support IdPs that are not able to support the values you list, then on receiving a SAML error you can try reissuing your SAML request with no <`RequestedAuthnContext>` element.

Use Cases for including the <RequestedAuthnContext> element

SP always requires MFA

This use case is most relevant if the SP operator knows that the IdP in question supports this profile. To require that all users must authenticate using MFA, a SAML authentication request should include:

```
<samlp:RequestedAuthnContext Comparison="exact">
     <saml:AuthnContextClassRef>
          http://id.incommon.org/assurance/mfa
     </saml:AuthnContextClassRef>
</samlp:RequestedAuthnContext>
```

That is, MFA is the (only) requested value.

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, and 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 one of the later uses case of 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.

SP prefers, but does not require, MFA

In some cases, an SP may *prefer* that users authenticate with MFA but is willing to accept non-MFA authentication. Some scenarios where this approach would 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.

IdP Known to support MFA and Base-level profiles

If the IdP is known to support the MFA and base-level profiles, and all that is of interest is whether or not MFA was performed, the SAML authentication request from the SP should include:

```
<samlp:RequestedAuthnContext Comparison="exact">
     <saml:AuthnContextClassRef>
          http://id.incommon.org/assurance/mfa
     </saml:AuthnContextClassRef>
     <saml:AuthnContextClassRef>
          http://id.incommon.org/assurance/base-level
     </saml:AuthnContextClassRef>
</samlp:RequestedAuthnContext>
```

This request snippet indicates that MFA profile is preferred (i.e., listed first) but that the IdP can “fall back” to any weaker methods that satisfy the base-level profile otherwise.

IdP not known to support MFA/Base-level profiles

If the SP does *not* know whether the IdP supports the two profiles, then we recommend that the SP request not just these two profiles, but also any other standard or common `<AuthnContextClassRef>` values that are acceptable and frequently encountered. A recommended request that should cover most of these use cases would include:

```
<samlp:RequestedAuthnContext Comparison="exact">
     <saml:AuthnContextClassRef>
          http://id.incommon.org/assurance/mfa
```

```
        </saml:AuthnContextClassRef>
        <saml:AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:X509
        </saml:AuthnContextClassRef>
        <saml:AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos
        </saml:AuthnContextClassRef>
        <saml:AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
        </saml:AuthnContextClassRef>
        <saml:AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:Password
        </saml:AuthnContextClassRef>
        <saml:AuthnContextClassRef>
            http://id.incommon.org/assurance/base-level
        </saml:AuthnContextClassRef>
    </samlp:RequestedAuthnContext>
```

The actual list of `AuthnContextClassRef` values to support is up to the SP, but this list is likely to address the majority of IdPs. To support arbitrary IdPs, it may still be necessary to respond to SAML errors by issuing a separate SAML request that includes no <`RequestedAuthnContext>` element.

Base-level profile is not sufficient

If MFA is desired, but the base level profile is not sufficient for an SP, the SP should list all acceptable values in the <`RequestedAuthnContext>` element. This would work similarly to the above use case, but without including the `http://id.incommon.org/assurance/base-level` value in the request.

SP Requires “Step Up” MFA

If a user was initially authenticated without MFA then depending on the identity of the user or the services the user is accessing, the SP may want to “elevate” the user’s authentication profile as a prerequisite to allowing further access. To do this, a new SAML authentication request must be generated that includes only `http://id.incommon.org/assurance/mfa`. This request would be equivalent to the requests generated under the “*SP always requires MFA*” section, above.