Usage Guidance

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

Overview

The *[link]*InCommon Base Level and *[link[*Multi-Factor Authentication (MFA) Profiles define requirements that allow Service Providers (SPs) to request that Identity Providers (IdPs) perform multi-factor 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 multi-factor 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.

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 NOT considered a second factor. Institutions are expected to provide safeguards to maintain the independence of their supported authentication factors

    - For example, software/virtual phones that are accessible using the enterprise password are not appropriate second factors.
    - 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 not secure. 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 is not intended to restrict this *initial* MFA factor registration practice.

SAML-Specific Guidance

The current profiles are specific to implementing within SAML. This section provides guidance on how to use the profiles.

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, so an SP should ideally determine whether a given IdP supports these profiles through some out of band mechanism. 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.

Use Cases

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 `http://id.incommon.org/assurance/mfa` as the (only) requested `<AuthnContextClassRef>` value.

***[Add:** Example AuthnRequest and selected software package config examples]*

Keep in mind that even if an IdP supports the MFA Profile, it can only successfully respond 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 the next use 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.

In these cases, the SAML authentication request from the SP should include both the `http://id.incommon.org/assurance/mfa` and the `http://id.incommon.org/assurance/base-level` `<AuthnContextClassRef>` values, with MFA being preferred (i.e., listed first). This will allow an IdP to use MFA if possible, but to “fall back” to weaker methods that satisfy the base level context otherwise. (If the SP has more specific authentication requirements than just “MFA was or was not used”, see the next section *“Base-level profile is not sufficient”* for further guidance)

***[Add:** Example AuthnRequest and selected software package config examples]*

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.

If the SP knows that the IdP supports the use of both `<AuthnContextClassRef>` values, the authentication process can end here. Otherwise, the SP may consider issuing a new authentication request that does not include any `<RequestedAuthnContext>` element, to allow IdPs that are not able to respond with either `<AuthnContextClassRef>` value to assert a different context (such as `urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport`).

Base-level profile is not sufficient

In the rare case that an application tracks or audits the actual authentication mechanism used by examining the asserted `<AuthnContextClassRef>` value, the non-MFA `http://id.incommon.org/assurance/base-level` context provides very little detail on the manner of user authentication.

If an application has such requirements, it may consider including specific `<AuthnContextClassRef>` values (such as `urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport`) in its initial request, to disallow the use of mechanisms it considers “too weak” (such as IP-based pseudo-authentication), rather than specifying the `http://id.incommon.org/assurance/base-level` value.

Alternatively, an initial request could be made specifying only the `http://id.incommon.org/assurance/mfa` context; if that results in an error then a new request could be issued without specifying anything, allowing an IdP to return its preferred `<AuthnContextClassRef>` value for the SP to examine and respond to appropriately.

SP Requires “Step Up” MFA

If a user was initially authenticated without MFA (e.g., using the `http://id.incommon.org/assurance/base-level` value rather than `http://id.incommon.org/assurance/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.

Responding to MFA Requests in SAML

IdPs Should Support both Profiles

We expect that many SPs that request the MFA context will also want to allow the user to authenticate if MFA is not available. We therefore recommend that any IdP that supports generating assertions that include the `http://id.incommon.org/assurance/mfa` value also be able to generate assertions that include `http://id.incommon.org/assurance/base-level` when requested.

MFA and IdP Session Length

The MFA profile places requirements on the authentication applied by the IdP; it does not prescribe any specific limitations on allowed session length, such as the period during which the IdP will respond to subsequent requests without re-authenticating the user.

Inexact Matching

SAML 2.0 includes the ability to request particular authentication mechanisms more indirectly (e.g., in plain language, “use something better than a password”). While this is a powerful and in some cases much more effective way of expressing requirements, it is also poorly supported, and requires significant configuration by deployers to work effectively.

While it is beneficial for an IdP with such capabilities to appropriately configure itself (e.g., allowing a request for “better” than `http://id.incommon.org/assurance/base-level` to be satisfied by `http://id.incommon.org/assurance/mfa`), the inconsistent support for this feature makes recommending it for use by SPs difficult.