Extended Enterprise MFA

Multifactor Authentication in the Extended Enterprise

There are known pockets of multifactor in use in higher ed today but more often than not multifactor is deployed at the service itself. This is entirely natural since the SP has a vested interest in its own security. This tendency is often reflected in the technology solutions themselves, which are primarily enterprise authentication technology solutions at best.

Given a particular authentication technology, the decision whether or not to deploy is a complicated one. Two specific questions tend to select the most flexible solutions:

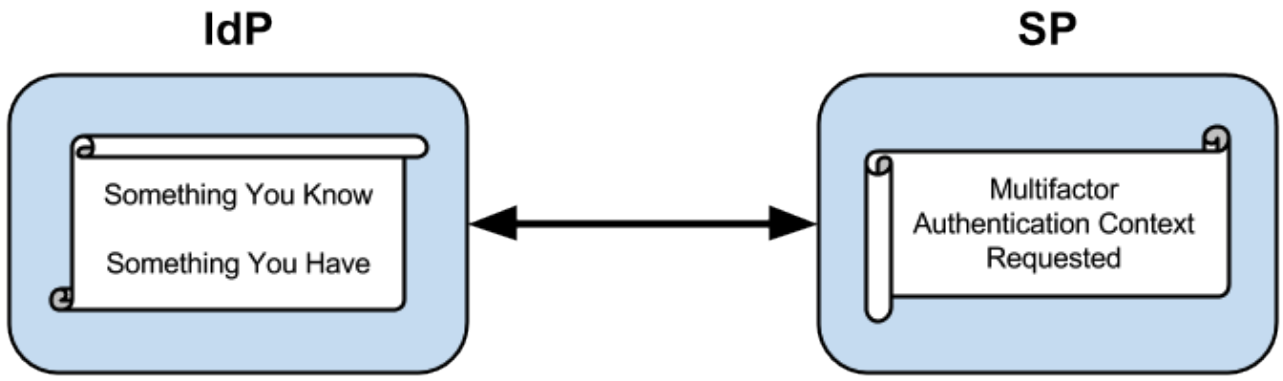
1. Does the authentication technology solution integrate with the enterprise SSO system?
2. Can the authentication technology solution be delivered “as-a-service?”

Let’s assume the answer to the first question is yes (an assumption on which the rest of this article is based). If the answer to the second question is no, we can conclude that the technology is limited to the enterprise, which means that one or the other of the SSO system or the authentication technology solution is limited to the enterprise. Regardless of the answers, this pair of questions tends to clarify the scope of the deployment so that there are no surprises down the road.

Enterprise Multifactor Authentication

In a federated scenario, the typical approach to multifactor authentication is to enable one or more additional factors in an existing enterprise IdP deployment that already authenticates its users with a username and password. This, too, is a natural thing to do, and it *does* have some advantages (to be discussed later) but this approach has at least one major drawback and, that is, it assumes all your users are managed by a single IdP. That’s a showstopper for any use case that touches the extended enterprise.

Patrick Harding refers to the notion of the extended enterprise in a recent article entitled Vanishing IT Security Boundaries Reappearing Disguised As Identity. Although the article itself focuses on identity technology, it’s the title that’s relevant here, which suggests that the limitations of the traditional security perimeter, typified by the firewall, give rise to the twin notions of *identity* and the *extended enterprise*. Neither is hardly a new concept but the notion highlights the limitations of what might be called the *enterprise deployment model of multifactor authentication*.



Multifactor Authentication Enterprise Deployment Model (single, multifactor IdP)

Integrating multifactor into the enterprise IdP is a definite improvement but it's still not good enough. How many times do you authenticate against your IdP in a typical work day? That's how many times you'll perform multifactor authentication, so your multifactor solution had better have darn good usability.

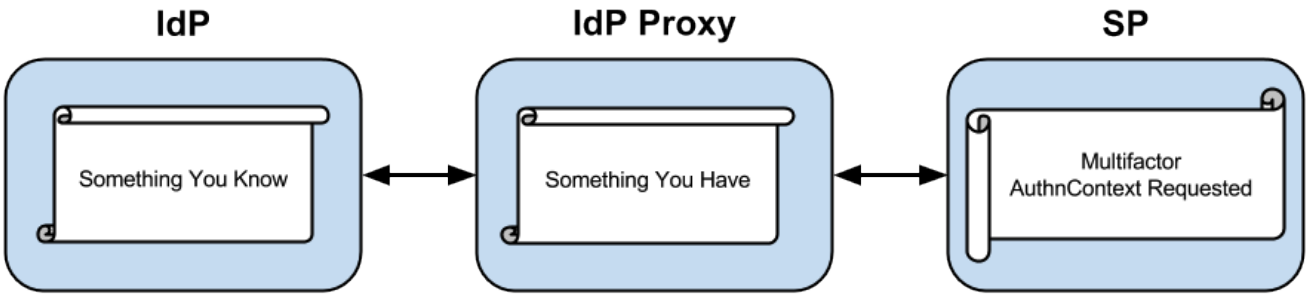
The fact is, unless your multifactor solution is passive, it *will* be less usable. The tendency will be to use the IdP's browser session to short-circuit an authentication request for multifactor. This is easier said, than done, however. Session management at the IdP was built with single factor authentication in mind. It'll be some time before SSO regains its footing at the IdP.

In the sections that follow, we show how to extend the enterprise deployment model in a couple of interesting and useful directions.

Distributed Multifactor Authentication

The core SAML spec alludes to a role called an *IdP Proxy*, apparently referring to a kind of two-faced IdP embedded in a chain of IdPs, each passing an assertion back to the end SP, perhaps consuming and re-issuing assertion content along the way. Since the SAML spec first appeared in 2005, the IdP Proxy has proven itself in production hub-and-spoke federations worldwide (particularly in the EU), but the idea of an IdP Proxy coupled with a multifactor authentication solution has only recently appeared. In particular, SURFnet began a project to deploy a Multifactor IdP Proxy in the SURFconext federation late in 2012. The *Multifactor IdP Proxy* enables what we call *distributed multifactor authentication*.

Consider the following use case. Suppose a high-value SP will *only* accept assertions from IdPs willing and able to assert a multifactor authentication context. Users, and therefore IdPs, are scattered all across the extended enterprise. Even if all the IdPs belong to the same federation, the trust problem becomes intractable beyond a mere handful of IdPs. A Multifactor IdP Proxy may be used to ease the transition to multifactor authentication across the relevant set of IdPs.



Distributed Multifactor Authentication
***Extended* Enterprise Deployment Model**
(multiple, single-factor IdPs)

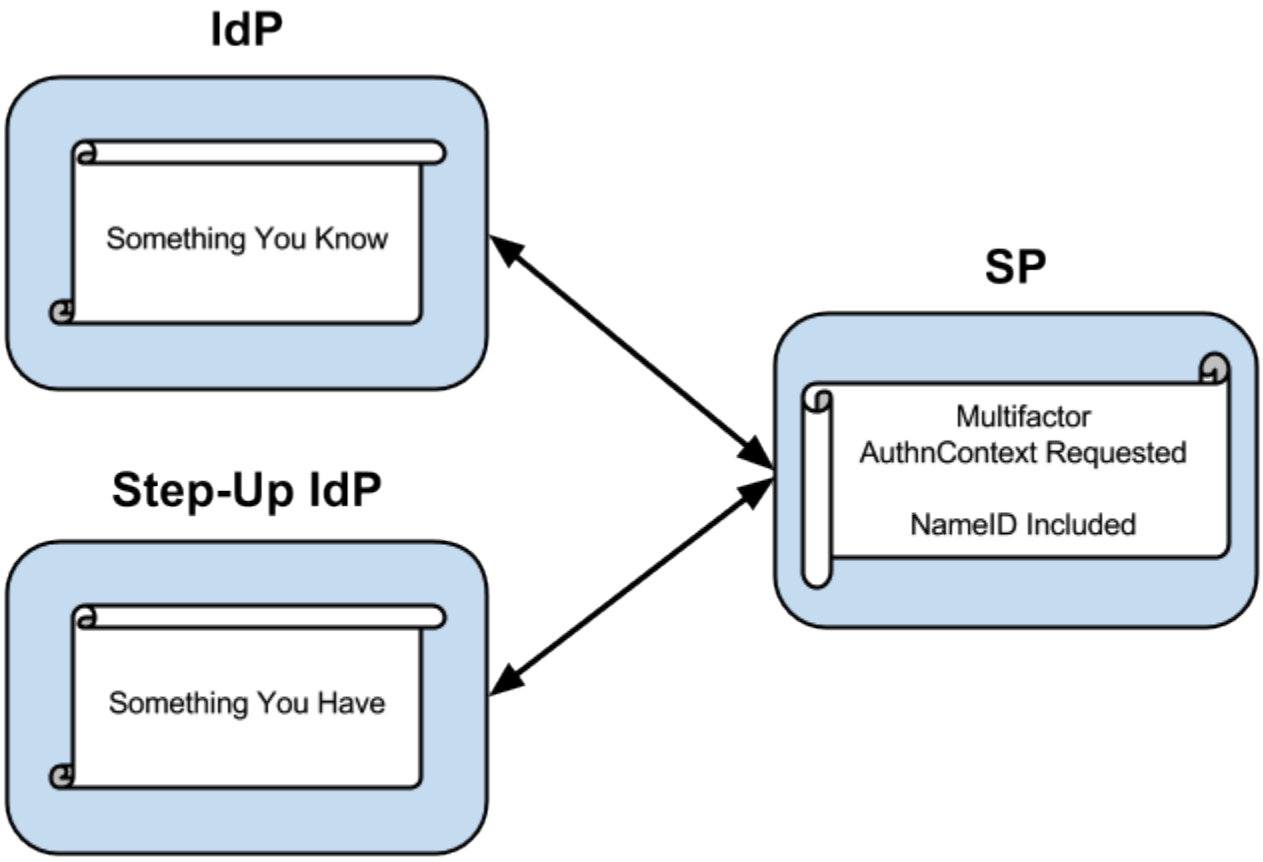
The Multifactor IdP Proxy performs two basic tasks: 1) it authenticates all browser users via some “what you have” second factor, and 2) it asserts a multifactor authentication context (plus whatever attributes are received from the end IdP). The Multifactor IdP Proxy we have in mind is extremely lightweight—it asserts no user attributes of its own and is therefore 100% stateless.

More importantly, the Multifactor IdP Proxy precludes the need for each IdP to support an MFA context. Instead each IdP can continue doing what it has been doing (assuming the existence of a sufficiently strong password in the first place). For its part, the SP need only trust one IdP partner to perform MFA, that is, the Multifactor IdP Proxy. If necessary, the SP can deploy the Multifactor IdP Proxy itself. In either case, the SP meets its high level of assurance requirements in relatively short order.

Step-Up Authentication

Consider the following general use case. A federated SP not knowing its users in advance accepts identity assertions from some set of IdPs. The SP trusts the assertion upon first use but—the network being the hostile place that it is—the SP wants assurances that the next time it sees an assertion of this identity, the user is in fact the same user that visited the site previously. The SP could work with its IdP partners to build such a trusted federated environment (or a federation could facilitate that trust on the SP’s behalf), but frankly that’s a lot of work, and moreover, *all* of the IdPs must authenticate their users with multiple factors before such a trusted environment may be realized. Consequently the SP takes the following action on its own behalf.

Instead of leveraging an IdP Proxy, or perhaps in addition to one, the SP leverages a distributed IdP for the purposes of *step-up authentication*. When a user arrives at the site with an identity assertion in hand, the SP at its discretion requires the user to authenticate again. This might happen immediately, as the user enters the front door of the site, or it might happen later, when the user performs an action that requires a higher level of assurance. The timing doesn’t matter. What matters is that the SP adds one or more additional factors to the user’s original authentication context, thereby elevating the trust surrounding the immediate transaction.



Step-Up Authentication
***Extended* Enterprise Deployment Model**
(multiple, single-factor IdPs)

Comparative Analysis

Each of the methods described above has at least one disadvantage:

- Enterprise MFA is relatively difficult to implement. In general, the login handlers associated with multifactor IdPs must implement complex logic. SSO will suffer unless session management at the IdP is up to the task.
- Login interfaces for multifactor IdPs tend to be more complicated (but done right, this can turn into an advantage).
- Some user attributes cross the IdP Proxy more easily than others. In particular, scoped attributes present significant issues when asserted across the IdP Proxy.
- The IdP Proxy represents a single point of failure *and* a single point of compromise. Compensating actions at the IdP Proxy may be required.
- Since the SP must assert user identity to the step-up IdP, a non-standard authentication request may be required. Moreover, depending on the authentication technology solution in use, the SP may have to sign the authentication request.

On the other hand, the *extended* enterprise deployment models have some distinct advantages:

- Since distributed MFA and step-up authentication incorporate single-factor IdPs, these solutions (compared to enterprise MFA IdPs) are significantly easier to implement and deploy.
- For both distributed MFA and step-up authentication, the trust requirements of IdPs throughout the extended enterprise are minimized. Indeed, an IdP can continue to do what it already does since higher levels of assurance at the IdP are not required (but can be utilized if present).
- If necessary, the SP can assume at least partial ownership of its own trust requirements. The deployment strength of the step-up IdP can be completely determined by the SP. No external multifactor dependencies need to exist.
- By separating the factors across multiple IdPs, flexible distributed computing options are possible. In particular, *Multifactor Authentication as-a-Service* (MFAaaS) becomes an option.

The latter pair of advantages may be significant for some use cases. Unlike the enterprise deployment model (which relies on a single IdP), the extended enterprise surrounding the SP may leverage multiple IdPs. Moreover, those IdPs can live in different security domains, which is what makes MFAaaS possible.