GridSP

Grid SP

By definition, a SAML SP is a participant in a browser profile. A SAML SP (such as the Shib SP) consumes two types of SAML responses:

1. an authentication response (POST or Artifact binding)
2. an attribute response (SOAP binding)

In the presence of attribute push, however, there is no attribute response since attributes are bundled with the authentication response. In either case, attributes are resolved based on a previous act of authentication at the IdP.

In contrast, a *Grid SP* does not participate in a browser profile, so a Grid SP is unlike a Shib SP. A Grid SP consumes SAML as follows:

1. a SAML response obtained as a result of standalone attribute query (SOAP binding)
2. one or more SAML assertions pushed as a result of X.509 authentication (X.509 or SOAP binding)

In the case of standalone attribute query, the Grid SP may be the principal (wielding an Attribute Query Client) or more likely the Grid SP is an entity acting on behalf of the principal (such as GridShibForGlobusToolkit).

In the other case, the Grid SP consumes pushed SAML assertions bound to X.509 certificates or SOAP messages. Like a pulled SAML response, the pushed assertions are used exclusively for access control (not authentication). However, a pushed assertion may include an `AuthenticationStatement` that describes a previous act of authentication, such as authentication to an online CA (like the GridShibCertificateAuthority) or a gateway (like a TeraGrid ScienceGateway).

Assertion Consumer Service

Here's one possible algorithm for GSI attribute-based authorization at the Grid SP:

1. Authenticate via X.509.
2. Permit access based on identity? If so, return ALLOW; otherwise continue.
3. Consume all pushed SAML assertions bound to the certificate at the well-known certificate extension.
4. Permit access based on pushed attributes? If so, return ALLOW; otherwise continue.
5. Pull attributes based on Subject Information Access (SIA) and Subject Alt Name extensions? If so, skip to step 9; otherwise continue.
6. Pull attributes based on bound SAML? If so, skip to step 9; otherwise continue.
7. Pull attributes based on Classic GridShib? If so, skip to step 9; otherwise continue.
8. Return DENY.
9. Query for attributes. Consume all returned SAML assertions.
10. Permit access based on combined pushed and pulled attributes? If so, return ALLOW; otherwise return DENY.

Here's more detail for steps 5, 6, and 7.

Pull attributes based on SIA and Subject Alt Name extensions

If there is an SIA extension, iterate over all values until a trusted IdP entityID is found. If there is a Subject Alt Name extension, iterate over all values until a SAML Subject is found. If both are found, pull attributes from the trusted IdP in the SIA extension based on the SAML Subject in the Subject Alt Extension.

Pull attributes based on bound SAML

If there are bound SAML assertions, iterate over all assertions until a trusted `Issuer` is found as follows:

```
while (more assertions and not found) {
  if (self-issued assertion) {
    if (contains nested assertion) {
      if (trusted Issuer) { found = true; }
    }
  } else {
    if (trusted Issuer) { found = true; }
  }
}
```

If a trusted issuer is found, pull attributes based on the Subject in a bound SAML assertion.

Pull attributes based on Classic GridShib

If a trusted IdP was found at step 5, or a trusted IdP is otherwise configured at the Grid SP, pull attributes based on the Subject DN in the certificate.