# Scoped User Identifiers

## Scoped User Identifiers

Recently a serious flaw was found in Office 365:

http://www.economyofmechanism.com/office365-authbypass.html

You should of course review the report and make your own determination but here's a spoiler: *The Office 365 application neglected to scope-check a user identifier, which allowed an arbitrary identity provider to assert any user identifier whatsoever and thereby gain unauthorized access to the application.*

Here are a few lessons learned from the Office 365 vulnerability.

> **Lesson Learned #1**
>
> An email address is not a user identifier.

The Office 365 incident revolved around the following SAML attribute:

**The IDPEmail attribute**

```
<saml2:Attribute FriendlyName="IDPEmail"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="IDPEmail">
   <saml2:AttributeValue
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:type="xsd:string">ikakavas@mymail.example.com</saml2:AttributeValue>
</saml2:Attribute>
```

From a SAML perspective, there are a number of problems with the `IDPEmail` attribute shown above but the point that matters most is: The `IDPEmail` attribute is not an email address at all. Its value may look like an email address, but in fact, the `IDPEmail` attribute corresponds to the User Principal Name (`userPrincipalName`) of the existing account of the user in Azure AD.

That's an important observation: The `IDPEmail` attribute is an identifier, and moreover, it is actually used by the Office 365 application for access control. That leads directly to our next observation.

> **Lesson Learned #2**
>
> All user identifiers must be scope-checked by the relying party.

Some identifiers are explicitly scoped. In other cases, an identifier is implicitly scoped to the issuer. In all cases, a user identifier must be scope-checked by the relying party.

### Scoped Attributes

Some user identifiers are explicitly scoped. For example, the so-called scoped attributes `eduPersonPrincipalName` and `eduPersonUniqueId` encode a "scope" directly in the attribute value:

**Two scoped attributes**

```
<saml2:Attribute FriendlyName="eduPersonPrincipalName"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6">
   <saml2:AttributeValue
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:type="xsd:string">cantor.2@osu.edu</saml2:AttributeValue>
</saml2:Attribute>

<saml2:Attribute FriendlyName="eduPersonUniqueId"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.13">
   <saml2:AttributeValue
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:type="xsd:string">83909230284@internet2.edu</saml2:AttributeValue>
</saml2:Attribute>
```

In the above examples, the “scope” is “osu.edu” and “internet2.edu,” respectively. Only one identity provider (IdP) is authorized to assert the scope in each case, namely, the Ohio State University IdP and the Internet2 IdP, respectively. A relying party must check the asserted scope just-in-time and be prepared to reject an identifier if it does not pass the scope check.

So how does the relying party know what “scope” an IdP is authorized to assert? Trusted metadata, of course! If you search the InCommon metadata aggregate you will find the following `<shibmd:Scope>` elements:

**Scopes in metadata**

```
<shibmd:Scope regexp="false">osu.edu</shibmd:Scope>
<shibmd:Scope regexp="false">internet2.edu</shibmd:Scope>
```

Each `<shibmd:Scope>` element is associated with the IdP authorized to assert that scope, namely, the Ohio State University IdP and the Internet2 IdP, respectively.

## Other Scoped Identifiers

The SAML2 Persistent NameID is explicitly scoped but in a completely different manner. Here is an example:

**SAML2 Persistent NameID**

```
<saml2:NameID
      Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
      NameQualifier="https://idp.example.org/shibboleth"
      SPNameQualifier="https://sp.example.org/shibboleth"
      >1234567890</saml2:NameID>
```

The SAML2 Persistent NameID is asserted by the IdP as a child element of the `<saml2:Subject>` element or the `eduPersonTargetedID` attribute. In either case, the `NameQualifier` XML attribute must be scope-checked, that is, it must match the actual issuer of the assertion. Otherwise the identifier should be discarded by default.

All of the examples thus far have been SAML attributes and identifiers but scope-checking is not a SAML issue per se. For instance, the OpenID Connect "sub" claim is a locally unique, non-reassigned identifier for the user (sub = subject). Here is an example of an ID token asserted by an OpenID provider:

**An OpenID token**

```
{
  "iss": "https://server.example.com",
  "sub": "24400320",
  "aud": "s6BhdRkqt3",
  "exp": 1311281970,
  "iat": 1311280970
}
```

By definition, the "sub" claim is implicitly scoped to the issuer. In practice, a relying party would store the pair (iss, sub) and subsequently scope-check an ID token by matching **both** the "iss" value **and** the "sub" value. This prevents an arbitrary OpenID provider from asserting an unauthorized "sub" value, intentionally or otherwise.

## Other Identifiers

At this point we’ve looked at the following identifiers:

- `eduPersonPrincipalName`
- `eduPersonUniqueId`
- SAML2 Persistent NameID
- `eduPersonTargetedID`
- OpenID Connect "sub" claim

These aren’t the only identifiers that need to be scope-checked, however. See the NameIdentifiers topic in the Shibboleth wiki for a more complete list.

All of the above identifiers have well-defined scope semantics. The `IDPEmail` attribute, OTOH, is ill-suited for cross-domain access control. While its name and value suggest an email address, the underlying identifier (`userPrincipalName`) has no documented scope semantics AFAIK.

**Lesson Learned #3**

If your federating software doesn't scope-check user identifiers, then that responsibility must be taken up by the application software (i.e., the application developer).

Note well: *The application developer must scope-check all identifiers asserted by untrusted 3rd parties*. This is especially true if the identifier is used for access control. Failure to do so may lead to major security holes like the one reported in Office 365.

Of course this assumes the application relies on scoped identifiers to begin with. In particular, an application should **never** rely on an email address to identify a user. An email address is not scoped. For instance, the email address trscavo@gmail.com may be legitimately asserted by **any** IdP. Conclusion: an email address makes a lousy user identifier

Explicitly scoped identifiers (such as `eduPersonPrincipalName`, `eduPersonUniqueId`, and SAML2 Persistent NameID) are preferred since smart middleware can do the work for you. For instance, the Shibboleth SP software scope-checks the above identifiers by default. In the case of the scoped attributes `eduPersonPrincipalName` and `eduPersonUniqueId`, Shibboleth checks the actual scope against the authorized scope(s) in metadata. In the case of the SAML2 Persistent NameID and `eduPersonTargetedID` (which are equivalent), the software checks the `NameQualifier` XML attribute against the actual issuer. In any case, if the software detects a mismatch, the user identifier is not accepted and thus not added to the user's session.

Other relying party software might do scope-checking, I don't know. If your software (SAML or otherwise) does this, please log into Confluence and add a comment to the end of this article.

---

https://twitter.com/trscavo