WebmailShibboleth

Shibboleth, Webmail, IMAP Proof-of-concept

In English

Requirements

- The webmail software must not see or use users' LDAP password, the IdP must not release even the hashed form of the password.
- IMAP must authenticate with username and password.
- If one has access to the webmail server, she must not have access to the IMAP on behalf of all users (she can however access to active users session).

Solution concepts

- The IdP and the IMAP server share an authentication database.
- With every webmail SP request the IdP generates a new password for that particular user and writes it to the database.
- The webmail SP receives this password with the attribute set and uses the username (e-mail address) and password to access the IMAP server.
- The IMAP server tries to authenticate against the database.
- In order to secure access, this password entry should contain an expiration time, which invalidates the password after the IdP session ends, so IMAP accepts only those users who has recently initiated active session at the IdP side.

Shibboleth IdP plugin

- We have developed an IdP plugin -attribute resolver- which can generate this short-lifetime password (called service token) for the user and write it to the database.
- Shibboleth IdP attribute resolver configuration is independent from the actual SP, so the plugin must check whether the current request came from an SP for which it needs to generate the token.
- The service token is sent in plain-text, so the Shibboleth attribute statement must be encrypted either by using artifact resolution over SSL/TLS or by using XML encryption with HTTP-Post.

IMAP configuration

- As we don't want to force the use of webmail, IMAP needs to use LDAP authentication as well.
- Most IMAP servers can be configured to use PAM, which can be configured to use arbitrary SQL tables for authentication and it also supports authentication chaining.

Webmail softwares

- For our proof-of-concept we have tried squirrelmail and roundcube with its HTTP-authentication plugin. If the SP is releasing the username and service token as PHP_AUTH_USER and PHP_AUTH_PW, this authentication module works out-of-the-box.

Magyarul

Koncepció

A webmail és a levelezőszerver (IMAP/POP3) együttes működését szeretnénk Shibbolizálni. A fő probléma abból áll, hogy a webmail az IMAP szerver felé felhasználónévvel és jelszóval autentikál. Az címtárban tárolt jelszót azonban nem adhatjuk ki az alkalmazásoknak, ráadásul legtöbb esetben ez egy hashelt jelszó.

A következő kritériumoknak kell teljesülniük:

- a webmail nem fér hozzá a felhasználó SSO jelszavához (még hashelt formátumban sem)
- az IMAP szerver jelszavas autentikációt használ, minden felhasználónak egyedi jelszava van
- a webmail feltörése esetén nem férhetnek hozzá az összes felhasználó levelezéséhez

A fenti kritériumokat az 'egyszer használatos', rövid lejáratú jelszó használata ('service token') kielégíti. Ebben az esetben az IdP minden egyes webmail bejelentkezéshez generál egy véletlen

jelszót, és ezt elmenti egy adatbázisban, (beállítva a jelszóhoz egy rövid lejárati időt) valamint elküldi a webmail SP-nek. A webmail ezen rövid lejáratú jelszó használatával autentikál az IMAP szerver felé.

A leírt gondolatmenet megvalósításához három komponens együttműködése szükséges:

- az IdP jelszót kell generáljon egy adatbázisba
- a webmailnek el kell érnie ezt a jelszót
- az IMAP szervernek a jelszóadatbázist kell használnia az autentikációra

Adatbázis struktúra

MySQL használata esetén a következő adatbázisstruktúra használható:

```
CREATE TABLE `service_tokens` (
  `uid` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `expiration` datetime NOT NULL,
  PRIMARY KEY  (`uid`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

IdP plugin

Az IdP plugin aktuális verziója a következő URL-ről tölthető le:

http://software.niif.hu/maven2/hu/niif/shibboleth-servicetoken/1.0. A `shibboleth-servicetoken-1.0.jar` -t illetve a megfelelő adatbázis drivert (MySQL esetén `mysql-connector.jar`) be kell másolni az `idp.war WEB-INF/lib` könyvtárába.

Az `attribute-resolver.xml` -ben a következő változtatásokat kell megtenni:

```
<!--xml semak megfelelo beallitasa -->
<AttributeResolver
  ....
  xmlns:niifconnector="urn:geant:niif.hu:dataconnector"
  xsi:schemaLocation="
      ....
      urn:geant:niif.hu:dataconnector classpath:/schema/servicetokendataconnector.xsd">

 <!-- onetimepassword definicio -->
 <resolver:AttributeDefinition id="serviceToken" xsi:type="Simple"
```

```
    xmlns="urn:mace:shibboleth:2.0:resolver:ad"
    sourceAttributeID="serviceToken">

    <resolver:Dependency ref="serviceTokenConnector" />

    <resolver:AttributeEncoder xsi:type="SAML2String"
        xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
        name="urn:geant:niif.hu:servicetoken" friendlyName="serviceToken" />
  </resolver:AttributeDefinition>

  <!-- uid definicio -->
  <resolver:AttributeDefinition id="uid" xsi:type="Simple"
xmlns="urn:mace:shibboleth:2.0:resolver:ad"
     sourceAttributeID="uid">
    <resolver:Dependency ref="myLDAP" />
    <resolver:AttributeEncoder xsi:type="SAML1String"
xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
        name="urn:mace:dir:attribute-def:uid" />
    <resolver:AttributeEncoder xsi:type="SAML2String"
xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
        name="urn:oid:0.9.2342.19200300.100.1.1" friendlyName="uid" />
  </resolver:AttributeDefinition>

  <!-- service token generalasa -->
  <resolver:DataConnector xsi:type="niifconnector:ServiceToken"
    id="serviceTokenConnector"
    sourceAttributeID="uid"
    generatedAttributeID="serviceToken"
    tableName="service_tokens"
    principalColumn="uid"
    passwordColumn="password"
    expirationColumn="expiration"
    passwordLifetime="XXXXXX"
    spEntityID="https://webmail.example.org/shibboleth" >

    <resolver:Dependency ref="myLDAP" />

    <dc:ApplicationManagedConnection
        jdbcDriver="com.mysql.jdbc.Driver"
        jdbcURL="jdbc:mysql://localhost:3306/shib_idp"
```

```
        jdbcUserName="*****"
        jdbcPassword="*****" />
  </resolver:DataConnector>
```

Fontos, hogy a `DataConnector` (másodpercekben értelmezett) `passwordLifetime` attribútumát jól állítsuk be, azaz hosszabb legyen, mint a webmail oldali SP session, de javasolt 24 óránál rövidebbre venni.

Az `attribute-filter.xml` -ben pedig ki kell engedni az `uid` és `serviceToken` attribútumokat a webmail sp-nek:

```
<AttributeFilterPolicy id="sendServiceTokenToWebmail">
 <PolicyRequirementRule xsi:type="basic:AttributeRequesterString"
  value="https://webmail.example.org/shibboleth" />

 <AttributeRule attributeID="uid">
   <PermitValueRule xsi:type="basic:ANY" />
 </AttributeRule>
 <AttributeRule attributeID="serviceToken">
   <PermitValueRule xsi:type="basic:ANY" />
 </AttributeRule>
</AttributeFilterPolicy>
```

IMAP konfiguráció (Cyrus imapd)

Imapd saját autentikáció

A Cyrus imapd-ben be kell állítani az SQL autentikációt. Ehhez a libsasl2-modules-sql debian csomagra is szükségünk lesz. A konfigurációt az imapd.conf-ban tehetjük meg:

```
sasl_mech_list: PLAIN
sasl_pwcheck_method: auxprop

sasl_auxprop_plugin: sql
sasl_sql_engine: mysql
sasl_sql_hostnames: localhost
sasl_sql_user: *****
sasl_sql_passwd: *****
```

```
sasl_sql_database: shib_idp
sasl_sql_select: SELECT password AS userPassword FROM service_tokens WHERE uid = '%u' AND
expiration > now()
```

Amennyiben az IMAP szervert nem TLS/SSL felett használjuk, ezek a beállítások nem biztonságosak!

Használat PAM-mal

PAM használata esetén távolítsuk el a libsasl2-modules-sql csomagot, mert felesleges logüzeneteket gyárt. Ezen kívül szükség van a saslauthd-re is, amit debian alatt a sasl2-bin csomagban találhatunk.

Az imapd.conf-ot a következőképp kell beállítani:

```
sasl_mech_list: PLAIN
sasl_pwcheck_method: saslauthd
```

A saslauthd-t az /etc/default/saslauthd fájlban kell engedélyeznünk:

```
START=yes
MECHANISMS="pam"
```

Az /etc/pam.d/imap fájlban kell az imap pam beállításokat megtenni. Adatbázis használatához a libpam-mysql csomag is szükséges. Ha az adatbázisos felhasználókhoz nincs lokális account, akkor a PAM 'account' metódusát permit-re kell állítani.

```
auth       sufficient   pam_ldap.so
auth       sufficient   pam_mysql.so use_first_pass user=****** passwd=****** \
  host=/var/run/mysqld/mysqld.sock db=shib_idp table=service_tokens usercolumn=uid
passwdcolumn=password \
  crypt=plain [where=expiration>now()]
auth       required     pam_deny.so
@include common-account
```

SP konfiguráció

A webmailt futtató webszerver Shibboleth konfigurációjában el kell fogadni a felhasználónevet és a jelszót az IdP-től. Az attribute-map.xml -hez a következő bejegyezéseket kell hozzáadni:

```
<Attribute name="urn:oid:0.9.2342.19200300.100.1.1" id="PHP_AUTH_USER"/>
<Attribute name="urn:geant:niif.hu:servicetoken" id="PHP_AUTH_PW"/>
```

Figyeljünk arra, hogy az attribute-policy.xml -ben ezeket az attribútumokat beengedjük! (Az alapértelmezett telepítés egy catch-all engedélyező szabályt tartalmaz, tehát az attribútum rendben meg fog jelenni.)

Webmail szoftverek konfigurációja

Squirrelmail

A Squirrelmailhez a Squirrelmail HTTP Authentication Plugin letöltésével és telepítésével elvégezhető az IdP által kiadott és az SP által láthatóvá tett felhasználónév és jelszó alapú bejelentkezés.

Roundcube

A HTTP Authentication Plugin telepítése után a plugin-ból el kell távolítani a következő sort:

```
public $task = 'login';
```

Változat #1
cziernorbert hozta létre 2026-04-14 13:22:14 CEST
cziernorbert frissítette 2026-04-14 13:24:17 CEST