IdP 3.x username normalization

Overview

The default behavior for LDAP is to perform a case-insensitive search for username. And the default behavior for the IdP is to accept the username in exactly the form as entered by the user. Which can lead to inconsistent behavior of the IdP if the user changes the form how the username is entered (all lower-case, all upper-case, various forms of mixed-case...).

The consequence is that some operations on the IdP (like storing a session) can fail, because the storage layer detects the key (username) is different from the previous case. This gets quickly complex, as the database search operations are also case insensitive (at least on MySQL in the default configuration) - returning the existing session for the user, but confusing the storage layer with a different key than what the storage layer requested.

The real impact is that besides logging errors and exceptions in the log file, the IdP can also break the login flow, displaying the exception with a cryptic error message to the user. (This happens when a user logs into a new service for the first time, and uses a case combination different from what the main key for the user is - what was used for the very first time the user used the IdP).

The IdP 3.x installation manual has been updated (as of 2017-03-01) to force the IdP to normalize to lower-case by setting `shibboleth.authn.Password.Lowercase` to `TRUE` in `/opt/shibboleth-idp/conf/authn/password-authn-config.xml`:

```
    <util:constant id="shibboleth.authn.Password.Lowercase" static-field="java.lang.Boolean.TRUE"/>
```

However, on IdPs that already have been operating in production, it is necessary to make database changes **at the same time** to change all stored sessions to lowercase.

The tables affected are:

- `tb_st` - where the sharedToken values are stored
- `shibpid` - where persistentId values are stored
- `StorageRecords` - where the IdP stores several different types of records

Corrective actions

The recommended sequence is:

1. Preliminary: install MySQL helper functions needed by the code below
2. Run the DB sanity checks recommended below to confirm the database does not contain duplicate entries
3. Stop the IdP (`service tomcat stop`) - outage starts here
4. Make a dump of the database.
5. Make the above change to the IdP configuration.
6. Run the corrective SQL code (below) to fix the database (normalize all usernames to lowercase).
7. Start the IdP again (`service tomcat start`) - outage ends here
8. Remove the MySQL helper functions

DB helper functions

The following functions define string transformations that are needed by the code below:

MySQL helper functions

```
CREATE FUNCTION fixValue (value LONGTEXT)
RETURNS LONGTEXT
RETURN CONCAT( LOWER(SUBSTRING_INDEX(SUBSTRING_INDEX(value, ':', 1), ':', -1)), SUBSTRING(value, LOCATE(':',
value), LENGTH(value)));

DELIMITER $$
CREATE FUNCTION fixList (list LONGTEXT)
RETURNS LONGTEXT
BEGIN
DECLARE todo LONGTEXT;
DECLARE done LONGTEXT;
SET done = '';
SET todo = list;
iter: LOOP
   IF NOT LOCATE(',',todo)>0 THEN
       LEAVE iter;
   END IF;
   SET done = CONCAT(done,fixValue(SUBSTR(todo,1,LOCATE(',',todo))));
   SET todo = SUBSTR(todo,LOCATE(',',todo)+1);
END LOOP iter;
RETURN CONCAT(done,fixValue(todo));
END$$
DELIMITER ;

DELIMITER $$
CREATE FUNCTION fixEntry (data LONGTEXT, entryName LONGTEXT)
RETURNS LONGTEXT
BEGIN
DECLARE searchFor LONGTEXT;
DECLARE foundAt INT;
DECLARE found2 INT;
SET searchFor = CONCAT('"', entryName, '":"');
SET foundAt = LOCATE(searchFor,data);
IF foundAt > 0 THEN
  SET found2 = LOCATE('"',data,foundAt + LENGTH(searchFor));
  IF found2 > 0 THEN
    RETURN CONCAT(  SUBSTR(data,1,foundAt + LENGTH(searchFor) - 1),  LOWER(SUBSTR(data,foundAt + LENGTH
(searchFor), found2 - foundAt - LENGTH(searchFor))), SUBSTR(data,found2));
  ELSE
    RETURN data;
  END IF;
ELSE
  RETURN data;
END IF;
END$$
DELIMITER ;
```

DB sanity checks

The following checks:

- Identify usernames that have not been used in lowercase
- Checks there are no duplicates in `tb_st - the two numbers returned by the second query should be the same`
- All transformations on StorageRecords would work - see the section below for details on what the transformations do

```
SELECT * FROM tb_st WHERE lower(uid) NOT LIKE BINARY uid;
SELECT COUNT(DISTINCT(uid)), COUNT(DISTINCT(lower(uid))) FROM tb_st;

SELECT localId, principalName, lower(principalName) FROM shibpid WHERE lower(principalName) NOT LIKE BINARY
principalName;

SELECT id, CONCAT( LOWER(SUBSTRING_INDEX(SUBSTRING_INDEX(id, ':', 1), ':', -1)), SUBSTRING(id, LOCATE(':', id),
LENGTH(id))) AS new_id FROM StorageRecords WHERE LOCATE(':',id)>0 HAVING id != BINARY new_id;
SELECT value, fixEntry(value,"nam") AS new_value FROM StorageRecords WHERE id ='_session' HAVING value !=
BINARY new_value;
SELECT value, fixEntry(fixEntry(value,"LDAPN"),"U") AS new_value FROM StorageRecords WHERE id ='authn/Password'
HAVING value != BINARY new_value;
SELECT value, fixList(value) AS new_value FROM StorageRecords WHERE id LIKE '%:_key_idx' HAVING value != BINARY
new_value;
```

DB corrective action

The following code:

- changes `tb_st.uid` and `shibpid.principalName` to lowercase
- in StorageRecords:
  - for records where the `id` is of the form `<username>:<keyword>`, turn the username part to lowercase
  - in records with `id "_session"`, change the "nam" field in the JSON data to lowercase
  - in records with `id "Authn/Password"`, change the "LDAPN" field in the JSON data to lowercase
  - in records with `id` of the form `<username>:_key_idx`, change the username to lowercase in each tuple inside the array this field holds

```
UPDATE tb_st SET uid = LOWER(uid);
UPDATE shibpid SET principalName = LOWER(principalName);

UPDATE StorageRecords SET id = CONCAT( LOWER(SUBSTRING_INDEX(SUBSTRING_INDEX(id, ':', 1), ':', -1)), SUBSTRING
(id, LOCATE(':', id), LENGTH(id)))  WHERE LOCATE(':',id)>0;
UPDATE StorageRecords SET value = fixEntry(value,"nam") WHERE id ='_session';
UPDATE StorageRecords SET value = fixEntry(fixEntry(value,"LDAPN"),"U") WHERE id ='authn/Password';
UPDATE StorageRecords SET value = fixList(value) WHERE id LIKE '%:_key_idx';
```

DB cleanup

Remove the helper functions once this operation is complete:

```
DROP FUNCTION fixValue;
DROP FUNCTION fixList;
DROP FUNCTION fixEntry;
```

Further information

- AAF github ticket: https://github.com/ausaccessfed/shibboleth-idp-installer/issues/191
- Discussion on shibboleth-users: http://shibboleth.1660669.n2.nabble.com/ERROR-org-opensaml-storage-impl-JPAStorageService-337-td7619097.html