# Foreman - Bug #15682

## User with invalid email address from LDAP auth source still able to log in

07/14/2016 05:36 AM - Dominic Cleal

| | | | |
|---|---|---|---|
| **Status:** | Closed | | |
| **Priority:** | Normal | | |
| **Assignee:** | Dominic Cleal | | |
| **Category:** | Authentication | | |
| **Target version:** | 1.13.0 | | |
| **Difficulty:** | | **Fixed in Releases:** | |
| **Triaged:** | No | **Found in Releases:** | Nightly |
| **Bugzilla link:** | | **Red Hat JIRA:** | |
| **Pull request:** | https://github.com/theforeman/foreman/pull/3650 | | |

**Description**

Since #14720, users with an invalid e-mail address stored in an LDAP auth source are now able to gain access to Foreman, and worse, the invalid e-mail address is stored in their user account.

1. log in with an LDAP account (with invalid mail) that doesn't yet exist in Foreman
2. the user edit form is shown and an "Error: Email is Required" message, blank email field
3. log out from the dropdown menu
4. log in again and you're taken to /hosts, not the user edit form
5. a "success" notification is shown stating that "Email address is invalid"

Logs from the **second** login attempt:

```
2016-07-14T10:32:43 [app] [I] Started POST "/users/login" for 127.0.0.1 at 2016-07-14 10:32:43 +01
00
2016-07-14T10:32:43 [app] [I] Processing by UsersController#login as HTML
2016-07-14T10:32:43 [app] [I]   Parameters: {"utf8"=>"□", "authenticity_token"=>"xfD9ymzwcrbLwMSVE
AibMLp7rk5iTK1kkl974csYplXZ4gE2eu7ibuz4f6CX+KjKwCfsuCi+gU/fJjDOXxn+mQ==", "login"=>{"login"=>"eins
tein", "password"=>"[FILTERED]"}, "commit"=>"»Login«"}
2016-07-14T10:32:43 [sql] [D]   ActiveRecord::SessionStore::Session Load (0.1ms)  SELECT  "session
s".* FROM "sessions" WHERE "sessions"."session_id" = ?  ORDER BY "sessions"."id" ASC LIMIT 1  [["s
ession_id", "d79b816f2236118ed649765fd24bb834"]]
2016-07-14T10:32:43 [app] [D] Setting current user thread-local variable to nil
2016-07-14T10:32:43 [sql] [D]    (0.1ms)  begin transaction
2016-07-14T10:32:43 [sql] [D]   SQL (0.3ms)  DELETE FROM "sessions" WHERE "sessions"."id" = ?  [["
id", 4427]]
2016-07-14T10:32:43 [sql] [D]    (31.4ms)  commit transaction
2016-07-14T10:32:43 [sql] [D]   ActiveRecord::SessionStore::Session Load (0.3ms)  SELECT  "session
s".* FROM "sessions" WHERE "sessions"."session_id" = ?  ORDER BY "sessions"."id" ASC LIMIT 1  [["s
ession_id", "cbcd7c57c534e6ead2616a7073bfb116"]]
2016-07-14T10:32:43 [sql] [D]   User Load (0.1ms)  SELECT  "users".* FROM "users" WHERE "users"."l
ower_login" = ? LIMIT 1  [["lower_login", "einstein"]]
2016-07-14T10:32:43 [sql] [D]   AuthSource Load (0.1ms)  SELECT  "auth_sources".* FROM "auth_sourc
es" WHERE "auth_sources"."id" = ? LIMIT 1  [["id", 3]]
2016-07-14T10:32:43 [sql] [D] LDAP auth with user einstein against LDAP-ldap.example.com
2016-07-14T10:32:43 [sql] [D] Retrieved LDAP Attributes for einstein: {:firstname=>"albert", :last
name=>"einstein", :mail=>"foo#bar", :login=>"einstein"}
2016-07-14T10:32:43 [sql] [D] Authenticated user einstein against LDAP-ldap.example.com authentica
tion source
2016-07-14T10:32:43 [sql] [D]   User Load (0.1ms)  SELECT  "users".* FROM "users" WHERE "users"."l
ower_login" = ? LIMIT 1  [["lower_login", "foreman_admin"]]
2016-07-14T10:32:43 [app] [D] Setting current user thread-local variable to foreman_admin
2016-07-14T10:32:43 [sql] [D] Updating user einstein attributes from auth source: [:firstname, :la
stname, :mail, :login]
2016-07-14T10:32:43 [sql] [D]    (0.1ms)  begin transaction
2016-07-14T10:32:43 [sql] [D]    (0.1ms)  SELECT COUNT(*) FROM "auth_sources" WHERE "auth_sources"
."type" IN ('AuthSourceHidden') AND "auth_sources"."id" = ?  [["id", 3]]
```

```
2016-07-14T10:32:43 [sql] [D]   User Exists (0.1ms)  SELECT  1 AS one FROM "users" WHERE (LOWER("u
sers"."login") = LOWER('einstein') AND "users"."id" != 66) LIMIT 1
2016-07-14T10:32:43 [sql] [D]   Usergroup Load (0.1ms)  SELECT "usergroups".* FROM "usergroups" IN
NER JOIN "cached_usergroup_members" ON "usergroups"."id" = "cached_usergroup_members"."usergroup_i
d" WHERE "cached_usergroup_members"."user_id" = ?  ORDER BY usergroups.name
  [["user_id", 66]]
2016-07-14T10:32:43 [sql] [D]   Usergroup Load (0.0ms)  SELECT "usergroups".* FROM "usergroups" WH
ERE "usergroups"."name" = ?  ORDER BY usergroups.name  [["name", "einstein"]]
2016-07-14T10:32:43 [sql] [D]    (0.1ms)  rollback transaction
2016-07-14T10:32:43 [app] [D] Setting current user thread-local variable to nil
2016-07-14T10:32:43 [sql] [D] Post-login processing for einstein
2016-07-14T10:32:43 [sql] [D]   User Load (0.1ms)  SELECT  "users".* FROM "users" WHERE "users"."l
ower_login" = ? LIMIT 1  [["lower_login", "foreman_admin"]]
2016-07-14T10:32:43 [app] [D] Setting current user thread-local variable to foreman_admin
2016-07-14T10:32:43 [sql] [D]    (0.1ms)  begin transaction
2016-07-14T10:32:43 [sql] [D]   User Load (0.1ms)  SELECT  "users".* FROM "users" WHERE "users"."i
d" = ?  ORDER BY firstname LIMIT 1  [["id", 66]]
2016-07-14T10:32:43 [sql] [D]    (0.1ms)  SELECT MAX("audits"."version") FROM "audits" WHERE "audi
ts"."auditable_id" = ? AND "audits"."auditable_type" = ?  [["auditable_id", 66], ["auditable_type"
, "User"]]
2016-07-14T10:32:43 [sql] [D]   SQL (0.3ms)  INSERT INTO "audits" ("action", "audited_changes", "a
uditable_id", "auditable_type", "user_id", "username", "auditable_name", "created_at", "version",
"request_uuid", "remote_address") VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
 [["action", "update"], ["audited_changes", "---\nmail:\n- \n- foo#bar\n"], ["auditable_id", 66],
["auditable_type", "User"], ["user_id", 22], ["username", "Anonymous Admin"], ["auditable_name", "
albert einstein"], ["created_at", "2016-07-14 09:32:43.228650"], ["version"
, 2], ["request_uuid", "74b5df17-664f-4e5c-854a-2284f732a1c0"], ["remote_address", "127.0.0.1"]]
2016-07-14T10:32:43 [sql] [D]   SQL (0.1ms)  UPDATE "users" SET "mail" = ?, "last_login_on" = ?, "
updated_at" = ? WHERE "users"."id" = ?  [["mail", "foo#bar"], ["last_login_on", "2016-07-14 09:32:
43.223482"], ["updated_at", "2016-07-14 09:32:43.224347"], ["id", 66]]
2016-07-14T10:32:43 [sql] [D]   Role Load (0.0ms)  SELECT  "roles".* FROM "roles" WHERE "roles"."b
uiltin" = ? LIMIT 1  [["builtin", 2]]
2016-07-14T10:32:43 [sql] [D]   Role Exists (0.1ms)  SELECT  1 AS one FROM "roles" INNER JOIN "use
r_roles" ON "roles"."id" = "user_roles"."role_id" WHERE "user_roles"."owner_id" = ? AND "user_role
s"."owner_type" = ? AND "roles"."id" = ? LIMIT 1  [["owner_id", 66], ["owne
r_type", "User"], ["id", 8]]
2016-07-14T10:32:43 [sql] [D]    (22.2ms)  commit transaction
2016-07-14T10:32:43 [sql] [D]   CACHE (0.0ms)  SELECT  "roles".* FROM "roles" WHERE "roles"."built
in" = ? LIMIT 1  [["builtin", 2]]
2016-07-14T10:32:43 [sql] [D]   CACHE (0.0ms)  SELECT  1 AS one FROM "roles" INNER JOIN "user_role
s" ON "roles"."id" = "user_roles"."role_id" WHERE "user_roles"."owner_id" = ? AND "user_roles"."ow
ner_type" = ? AND "roles"."id" = ? LIMIT 1  [["owner_id", 66], ["owner_type
", "User"], ["id", 8]]
2016-07-14T10:32:43 [app] [D] Setting current user thread-local variable to nil
2016-07-14T10:32:43 [app] [D] Setting current user thread-local variable to einstein
2016-07-14T10:32:43 [app] [I] Redirected to http://0.0.0.0:3000/hosts
2016-07-14T10:32:43 [app] [I] Completed 302 Found in 88ms (ActiveRecord: 55.9ms)
```

These show the invalid mail is being updated during the post-login transaction. I think this comes from the User#post_successful_login method that updates last_login_on without validation, so the invalid mail that was previously assigned to the user account during the update attributes code in User.try_to_login also gets committed, despite it being invalid.

**Related issues:**

| | | |
|---|---|---|
| Related to Foreman - Bug #14720: User not shown error message when invalid em... | **Closed** | **04/19/2016** |

### Associated revisions

**Revision 29ff6661 - 07/20/2016 07:03 AM - Dominic Cleal**

fixes #15682 - don't save invalid attributes at login from LDAP

When a user logs in and their last_login_on attribute is updated, bypass saving the whole model which may contain invalid, unpersisted data.

Also fixes the warning about invalid synced attributes to show during first user login and not only subsequent logins, by restoring the invalid attributes to the model after saving it.

## History

#### #1 - 07/14/2016 05:37 AM - Dominic Cleal

*- Related to Bug #14720: User not shown error message when invalid email address copied from external auth source  added*

#### #2 - 07/14/2016 08:50 AM - The Foreman Bot

*- Status changed from Assigned to Ready For Testing*

*- Pull request https://github.com/theforeman/foreman/pull/3650 added*

#### #3 - 07/20/2016 08:01 AM - Dominic Cleal

*- Status changed from Ready For Testing to Closed*

*- % Done changed from 0 to 100*

Applied in changeset 29ff66614d2fefab3dcac452b5a2c46d3d3ffa5b.

#### #4 - 05/24/2020 06:45 PM - The Foreman Bot

*- Pull request https://github.com/theforeman/foreman/pull/7688 added*

#### #5 - 05/24/2020 06:51 PM - Ohad Levy

*- Triaged set to No*

*- Pull request deleted (https://github.com/theforeman/foreman/pull/7688)*