Foreman - Bug #4457

CVE-2014-0090 - Session fixation, new session IDs are not generated on login

02/26/2014 08:19 AM - Dominic Cleal

Status:	Closed		
Priority:	Urgent		
Assignee:	Dominic Cleal		
Category:	Security		
Target version:	1.4.2		
Difficulty:		**Fixed in Releases:**	
Triaged:		**Found in Releases:**	
Bugzilla link:		**Red Hat JIRA:**	
Pull request:			

Description

Description
Since new session id is not generated every time users log in, authentication can be bypassed through session fixation attacks in the situation where attackers are able to fixate another user's session id. Once users log in with the session id, attackers could also access the whole site with the user's privilege.

Severity: Medium

Affected URLs
http://$foreman/users/login

Steps
At host A, get a new session_id by accessing /users/login with any existing cookie removed.
At host B, access /users/login through http proxy. Intercept a request and delete Cookie header if exists. Intercept its response and modify _session_id in Set-cookie header with the one got in host A.
At host B, access /users/login and verify if the injected _session_id is using in Cookie header.
At host B, log in with admin(or any user) account.
At Host A, verify if the session is considered as authenticated.

Result
User at host A can access the application bypassing authentication

Remedy advice
The session ID should be always changed when users log in.

Reference
https://www.owasp.org/index.php/Session_fixation

Related issues:

Related to Foreman - Refactor #23875: Remove login doesn't escalate privilege...	**Closed**	**06/11/2018**

Associated revisions

Revision cfa4b526 - 03/24/2014 08:32 AM - Dominic Cleal

fixes #4457 - Session fixation, new session IDs are not generated on login (CVE-2014-0090)

Revision 7c67cfe4 - 03/24/2014 08:58 AM - Dominic Cleal

fixes #4457 - Session fixation, new session IDs are not generated on login (CVE-2014-0090)

(cherry picked from commit cfa4b52638173b9cf77ee1a5fd0c3a273f875209)

Conflicts:
test/functional/users_controller_test.rb

History

#1 - 02/26/2014 10:46 AM - Dominic Cleal

- Subject changed from Session fixation, new session IDs are not generated on login to CVE-2014-0090 - Session fixation, new session IDs are not

generated on login

- Description updated

#2 - 03/03/2014 01:29 PM - Dominic Cleal

- Target version set to 1.9.0

#3 - 03/04/2014 02:04 PM - Dominic Cleal

- Due date set to 03/18/2014

#4 - 03/09/2014 03:23 PM - Joseph Magen

- Status changed from New to Ready For Testing

- Assignee set to Joseph Magen

patch sent by email

#5 - 03/10/2014 08:12 AM - Dominic Cleal

- File 0001-fixes-4457-Session-fixation-new-session-IDs-are-not-.patch added

Please just upload patches here, thanks.

#6 - 03/10/2014 11:05 AM - Dominic Cleal

Review comments of v1 patch:

1. Should this only be in the request.post? branch, so we're not cycling sessions for every render of the login page?
2. This should also be in extlogin.
3. This introduces a regression with saved items within the session (see session_expiry in app controller), they're deleted as the session is reset.
4. Are the API controllers vulnerable too, since they also use the session?

#7 - 03/17/2014 08:33 AM - Joseph Magen

- File 0002-fixes-4457-Session-fixation-new-session-IDs-are-not-.patch added

changes in v2 patch attached

1. moved to request.post?
2. added to extlogin.
3. saved original_uri. good catch
4. API not effected. It doesn't create a new session session. Either the authentication must be part of the request or the it checks if there is an existing user session (by UI) already exists.
https://github.com/theforeman/foreman/blob/develop/app/controllers/concerns/foreman/controller/authentication.rb#L19

#8 - 03/17/2014 09:08 AM - Dominic Cleal

3. the session_expiry handler also saves the current taxonomy - maybe we should put this into a shared method so we don't have discrepancies

4. my understanding is that not creating new session IDs **is** the issue here. The security bug is that we take the session ID from the request, then escalate its privileges by assigning a current user so an attacker with a copy of the session ID (or who planted it into a user's request) gets escalated privileges.

The solution in the non-API controllers is to always generate new session IDs when we authenticate and escalate privileges. Since we do this on every request in the API, I'm not sure what the equivalent would be. Generating a new session on every request sounds expensive. Can we disable session handling in these controllers, or stop the authentication concern from storing escalated user details in the session when inside API controllers?

#9 - 03/17/2014 11:38 AM - Dominic Cleal

4. Ah, perhaps it's already the case that the API controllers don't touch the session:
https://github.com/theforeman/foreman/blob/0f7d219a4a65cd795eecd05117b08511d9025de2/test/functional/api/base_controller_subclass_test.rb#L44-L47

EDIT:
https://github.com/theforeman/foreman/blob/355bce36288ec6cd9f5a66b656f8a84158f2a8e1/app/controllers/concerns/foreman/controller/authentication.rb#L19

The only minor issue is that the base controller test doesn't test in the case where login=true, only login=false.

#10 - 03/17/2014 05:07 PM - Joseph Magen

- File 0003-fixes-4457-Session-fixation-new-session-IDs-are-not-.patch added

updated to include location_id, organization_id is saved session.

The API doesn't "login". Either the it users the session from a currently logged in user, or it needs to pass authentication credentials in the request. So, I don't see how the API touches this bug.

#11 - 03/17/2014 05:44 PM - Dominic Cleal

- Status changed from Ready For Testing to Pending

ACK, tests well. Thanks Joseph! I think we can release this tomorrow.

I agree that the API's unaffected, but the API does use the same Foreman::Authorization concern to process basic authentication as the main application so I thought that was vulnerable. It sets User.current, but it doesn't change the session because api_request? is true (see link above).

#12 - 03/17/2014 06:33 PM - Dominic Cleal

- Status changed from Pending to Ready For Testing

I take that back, the tests are failing (test/functional/users_controller_test.rb).

The test failures expose a bigger problem, which is that under SSO authentication (which is triggered under any request protected by the require_login filter), the issue is still present. The authenticate method in the Foreman::Controller::Authentication concern will set session[:user], but the session isn't reset first.

#13 - 03/18/2014 08:21 AM - Dominic Cleal

- Due date changed from 03/18/2014 to 03/20/2014

#14 - 03/18/2014 09:10 AM - Joseph Magen

Are we referring to this line in the code regarding SSO
https://github.com/theforeman/foreman/blob/develop/app/controllers/concerns/foreman/controller/authentication.rb#L64

This calls authenticate! for each provider.

I haven't worked with the SSO code, so I think it will be more efficient if Marek can take a look at this issue (as I think he wrote the SSO code). I don't have apache installed locally to test it.

#15 - 03/18/2014 09:21 AM - Dominic Cleal

- Assignee changed from Joseph Magen to Dominic Cleal

#16 - 03/18/2014 03:48 PM - Dominic Cleal

- File 0001-fixes-4457-Session-fixation-new-session-IDs-are-not-.patch added

Attached a new patch for review, which resets the session when the user is authenticated via UsersController#login and also via Foreman::Controller::Authentication (used for SSO), but not when it's an API request (where we don't store the user in the session).

#17 - 03/18/2014 03:51 PM - Dominic Cleal

- File deleted (0001-fixes-4457-Session-fixation-new-session-IDs-are-not-.patch)

#18 - 03/18/2014 03:52 PM - Dominic Cleal

- File 0001-fixes-4457-Session-fixation-new-session-IDs-are-not-.patch added

Apologies, missed a file from the commit.

#19 - 03/19/2014 09:48 AM - Joseph Magen

Dominic, the session[:location_id] and session[:organization_id] are saving correctly when I test it on my local machine.

it seems this line is not working correctly
backup_session_content { expire_session }

#20 - 03/19/2014 01:18 PM - Dominic Cleal

As per our IRC conversation, this works for me. When testing expiry, set the idle_timeout setting to '1' and wait a minute for the session to expire. Deleting your session ID isn't the same as a time-based expiry - it's only on a time-based expiry that we know the user is still the same one, and are able to restore their context.

#21 - 03/20/2014 04:52 PM - Joseph Magen

Dom, are you waiting on me for any action items? I still think that location_id and organizatin_id can be saved in a similar way to session[:original_uri]

#22 - 03/20/2014 05:23 PM - Greg Sutcliffe

:+1:

I've reproduced the issue locally and tested Dominic's patch, which correctly fixes the issue. I've also tested that the session expiry correctly stores the user's current taxonomy scope, so it's restored when logging back in.

Good to merge from my perspective.

#23 - 03/20/2014 05:25 PM - Dominic Cleal

Yes, I don't think you're testing the right thing, and even before this patch, what you're testing wouldn't work. On session expiry (that is, the time limit by the idle_timeout setting) then the current context is preserved with this patch.

If you delete your session ID then you're testing something else. We can only restore original_uri in that case because the browser is making a new request for it, while the browser doesn't provide anything in the request about its taxonomy context. We could store this in a cookie instead, but this is unrelated to the session fixation issue and I don't think I've caused a regression.

I do however need an ack from another maintainer to proceed, thanks.

#24 - 03/20/2014 08:41 PM - Joseph Magen

Dom, thanks for the explanation.

#25 - 03/21/2014 05:27 PM - Dominic Cleal

- Status changed from Ready For Testing to Pending

#26 - 03/24/2014 08:56 AM - Dominic Cleal

- Private changed from Yes to No

#27 - 03/24/2014 09:31 AM - Dominic Cleal

- Status changed from Pending to Closed

- % Done changed from 0 to 100

Applied in changeset cfa4b52638173b9cf77ee1a5fd0c3a273f875209.

#28 - 03/25/2014 09:42 AM - Dominic Cleal

- Description updated

#29 - 06/11/2018 09:52 AM - Lukas Zapletal

- Related to Refactor #23875: Remove login doesn't escalate privileges test added

Files

0001-fixes-4457-Session-fixation-new-session-IDs-are-not-.patch	1000 Bytes	03/10/2014	Dominic Cleal
0002-fixes-4457-Session-fixation-new-session-IDs-are-not-.patch	1.58 KB	03/17/2014	Joseph Magen
0003-fixes-4457-Session-fixation-new-session-IDs-are-not-.patch	1.69 KB	03/17/2014	Joseph Magen
0001-fixes-4457-Session-fixation-new-session-IDs-are-not-.patch	14.9 KB	03/18/2014	Dominic Cleal