mgr - Feature #40329

Feature # 47765 (New): mgr/dashboard: security improvements

mgr/dashboard: It should be possible to set an expiration date for the user password

06/13/2019 12:29 PM - Tiago Melo

Status:	Closed	**% Done:**	100%
Priority:	Normal		
Assignee:	Tatjana Dehler		
Category:	dashboard/usermgmt		
Target version:	v15.0.0		
Source:		**Reviewed:**	
Tags:		**Affected Versions:**	
Backport:		**Pull request ID:**	30939

Description

As as admin I should be able to set a TTL for the password to expire.
It should be a cluster wide configuration.
p.e.: every user should change his password every 3 months.

Further questions:

- admin password expiry: Should it be possible to set an expiry date for the admin password as well? Or only if there is at least another admin account? If it should not be possible to set expiry date prevent the user from doing so.
- disabled users password expiry: Should it be possible to set/have an expiry date for disabled users?
- 'ac_user_create_cmd' requires timestamp as 'pwd_expiry_date': The function (ac_user_create_cmd) to create a user on the command line requires a timestamp as 'pwd_expiry_date' at the moment. Do we want to keep it or change the behavior here?
- recalculate password expiry date: issue https://tracker.ceph.com/issues/40329 introduces a default expiry span (USER_PWD_DEFAULT_EXPIRY_SPAN) for the user passwords and adds a password expiry date field (pwd_expiry_date) to the User class. If the administrator edits the USER_PWD_DEFAULT_EXPIRY_SPAN variable the password expiry dates need to be re-calculated.
- update password expiry date (which is set manually): If the 'USER_PWD_DEFAULT_EXPIRY_SPAN' is set and the user changes the password, it's easy to update the expiry date to the next date. But what happens if 'USER_PWD_DEFAULT_EXPIRY_SPAN' is not set and the password expiry date was entered manually?

Subtasks:

Feature # 40814: mgr/dashboard: Allow to set individual password expiry dates	**Closed**
Feature # 40816: mgr/dashboard: Recalculate password expiry date	**Closed**
Feature # 42340: mgr/dashboard: admin password expiry	**Closed**
Feature # 42342: mgr/dashboard: disabled users password expiry	**Closed**
Feature # 42343: mgr/dashboard: 'ac_user_create_cmd' requires timestamp as 'pwd_expiry_...	**Closed**

Related issues:

Related to mgr - Feature #40248: mgr/dashboard: As a user, I want to change m...	**Closed**
Related to mgr - Feature #25229: mgr/dashboard: Provide user enable/disable c...	**Closed**
Related to mgr - Feature #24655: mgr/dashboard: Enforce password change upon ...	**Closed**
Related to mgr - Feature #25232: mgr/dashboard: Support minimum password comp...	**Closed**
Related to mgr - Feature #39999: mgr/dashboard: Prevent brute-force/dictionar...	**New**
Related to mgr - Fix #40328: mgr/dashboard: Permanent notifications instead o...	**New**
Related to mgr - Bug #43431: mgr/dashboard: test_create_with_default_expirati...	**Resolved**
Copied to mgr - Backport #46837: nautilus: mgr/dashboard: user management imp...	**Rejected**

History

#1 - 07/03/2019 09:34 AM - Tatjana Dehler

- Assignee set to Tatjana Dehler

#2 - 07/04/2019 12:52 PM - Tatjana Dehler

- Description updated

My current idea:

1. How to configure the expiration date (admin perspective)
   A new setting *USER_PWD_EXPIRY_SPAN* will be added to *settings.py*. By default the value will be set to 0, which means the user passwords are never going to expire. The admin can configure this setting on command line (as a first step, would be great to integrate it into a configuration page at a later point of time) and set the amount of days to expire the password.
2. How to check if the password is expired (internal implementation)
   A new attribute *lastPwdUpdate* (unix timestamp) will be added to the *user* class in *access_control.py*. When a user logs in and the *USER_PWD_EXPIRY_SPAN* is defined, the current time will be compared with the *lastPwdUpdate* value. If (*current_time* - *lastPwdUpdate*) < *USER_PWD_EXPIRY_SPAN* nothing happens. If (*current_time* - *lastPwdUpdate*) >= *USER_PWD_EXPIRY_SPAN* the user will be asked to set a new password. After that *lastPwdUpdate* will be set to the current time. It would be great if the implementation of https://tracker.ceph.com/issues/24655 can used here.
3. How the user gets notified about it
   When a user tries to log in, he gets notified about the expired password on the login page. He'll be asked to set a new password which will also update the *lastPasswordUpdate* field to the current time. It would be great if the implementation of https://tracker.ceph.com/issues/24655 can used here as well.

Any further ideas/comments?

#3 - 07/04/2019 01:57 PM - Ricardo Dias

Tatjana Dehler wrote:

> My current idea:
>
> 1. How to configure the expiration date (admin perspective)
>   A new setting *USER_PWD_EXPIRY_SPAN* will be added to *settings.py*. By default the value will be set to None, which means the user passwords are never going to expire. The admin can configure this setting on command line (as a first step, would be great to integrate it into a configuration page at a later point of time) and set the amount of days to expire the password.

This means that the expiration span will be the same for all users. I'm not sure about the requirements for this feature, but we could easily define a different expiration span for each user by using a "pwdExpirySpan" attribute in the user profile. We could still have a *USER_PWD_DEFAULT_EXPIRY_SPAN* as a default value, if the admin wants the same for every user.

> 1. How to check if the password is expired (internal implementation)
>   A new attribute *lastPwdUpdate* (unix timestamp) will be added to the *user* class in *access_control.py*. When a user logs in and the *USER_PWD_EXPIRY_SPAN* is defined, the current time will be compared with the *lastPwdUpdate* value. If (*current_time* - *lastPwdUpdate*) < *USER_PWD_EXPIRY_SPAN* nothing happens.

I would use a "pwdExpirationDate" that would be set upon password change, and this allows to do a simpler comparison on login: _current_time < pwdExpirationDate

> If (*current_time* - *lastPwdUpdate*) >= *USER_PWD_EXPIRY_SPAN* the user will be asked to set a new password. After that *lastPwdUpdate* will be set to the current time. It would be great if the implementation of https://tracker.ceph.com/issues/24655 can used here.

I think we cannot request the user to change the password when the password expires because of security implications. Imagine the case of an attack that steals a list of active passwords. Then if the user did not change the password before the expiration data, the attacker can do the login in the system with the expired password, and the system will allow the attacker to change the password.

I think the correct approach here is to lock down the account when the password expires. Therefore the user should change the password before the current password expires.
If the user does not change the password in time, it must request the Administrator to enable the account with a new password.

> > 1. How the user gets notified about it
> > When a user tries to log in, he gets notified about the expired password on the login page. He'll be asked to set a new password which will also update the *lastPasswordUpdate* field to the current time. It would be great if the implementation of https://tracker.ceph.com/issues/24655 can used here as well.
>
> Any further ideas/comments?

I think we should issue a warning upon login a few (configurable) days before the password expires so that the user has the time to change the password.

Also, this feature can only be implemented after https://tracker.ceph.com/issues/40248 is resolved, otherwise users will not be able to change their passwords.

#4 - 07/04/2019 02:03 PM - Patrick Seidensal

Ricardo Dias wrote:

> Tatjana Dehler wrote:
>
> > My current idea:
> >
> > 1. How to configure the expiration date (admin perspective)
> > A new setting *USER_PWD_EXPIRY_SPAN* will be added to *settings.py*. By default the value will be set to None, which means the user passwords are never going to expire. The admin can configure this setting on command line (as a first step, would be great to integrate it into a configuration page at a later point of time) and set the amount of days to expire the password.
>
> This means that the expiration span will be the same for all users. I'm not sure about the requirements for this feature, but we could easily define a different expiration span for each user by using a "pwdExpirySpan" attribute in the user profile. We could still have a *USER_PWD_DEFAULT_EXPIRY_SPAN* as a default value, if the admin wants the same for every user.

I cannot think of a scenario where some users need to change their passwords more frequently than others. I'd postpone this implementation until it's actually required or at least clear, that some customers require such a feature.

> > 1. How to check if the password is expired (internal implementation)
> > A new attribute *lastPwdUpdate* (unix timestamp) will be added to the *user* class in *access_control.py*. When a user logs in and the *USER_PWD_EXPIRY_SPAN* is defined, the current time will be compared with the *lastPwdUpdate* value. If (*current_time - lastPwdUpdate*) < *USER_PWD_EXPIRY_SPAN* nothing happens.

> I would use a "pwdExpirationDate" that would be set upon password change, and this allows to do a simpler comparison on login: _current_time < pwdExpirationDate
>
>> If (*current_time* - *lastPwdUpdate*) >= *USER_PWD_EXPIRY_SPAN* the user will be asked to set a new password. After that *lastPwdUpdate* will be set to the current time. It would be great if the implementation of https://tracker.ceph.com/issues/24655 can used here.

> I think we cannot request the user to change the password when the password expires because of security implications. Imagine the case of an attack that steals a list of active passwords. Then if the user did not change the password before the expiration data, the attacker can do the login in the system with the expired password, and the system will allow the attacker to change the password.
>
> I think the correct approach here is to lock down the account when the password expires. Therefore the user should change the password before the current password expires.
> If the user does not change the password in time, it must request the Administrator to enable the account with a new password.
>
>> 1. How the user gets notified about it
>>   When a user tries to log in, he gets notified about the expired password on the login page. He'll be asked to set a new password which will also update the *lastPasswordUpdate* field to the current time. It would be great if the implementation of https://tracker.ceph.com/issues/24655 can used here as well.
>>
>> Any further ideas/comments?

> I think we should issue a warning upon login a few (configurable) days before the password expires so that the user has the time to change the password.
>
> Also, this feature can only be implemented after https://tracker.ceph.com/issues/40248 is resolved, otherwise users will not be able to change their passwords.

I like the rest of the proposal!

#5 - 07/04/2019 02:08 PM - Tatjana Dehler

Ricardo Dias wrote:

> Tatjana Dehler wrote:
>
> > My current idea:
> >
> > 1. How to configure the expiration date (admin perspective)
> > A new setting *USER_PWD_EXPIRY_SPAN* will be added to *settings.py*. By default the value will be set to None, which means the user passwords are never going to expire. The admin can configure this setting on command line (as a first step, would be great to integrate it into a configuration page at a later point of time) and set the amount of days to expire the password.
>
> This means that the expiration span will be the same for all users. I'm not sure about the requirements for this feature, but we could easily define a different expiration span for each user by using a "pwdExpirySpan" attribute in the user profile. We could still have a *USER_PWD_DEFAULT_EXPIRY_SPAN* as a default value, if the admin wants the same for every user.

Yes, fine with me.

> > 1. How to check if the password is expired (internal implementation)
> > A new attribute *lastPwdUpdate* (unix timestamp) will be added to the *user* class in *access_control.py*. When a user logs in and the *USER_PWD_EXPIRY_SPAN* is defined, the current time will be compared with the *lastPwdUpdate* value. If (*current_time* - *lastPwdUpdate*) < *USER_PWD_EXPIRY_SPAN* nothing happens.
>
> I would use a "pwdExpirationDate" that would be set upon password change, and this allows to do a simpler comparison on login: _current_time < pwdExpirationDate

Also fine with me.

> > If (*current_time* - *lastPwdUpdate*) >= *USER_PWD_EXPIRY_SPAN* the user will be asked to set a new password. After that *lastPwdUpdate* will be set to the current time. It would be great if the implementation of https://tracker.ceph.com/issues/24655 can used here.
>
> I think we cannot request the user to change the password when the password expires because of security implications. Imagine the case of an attack that steals a list of active passwords. Then if the user did not change the password before the expiration data, the attacker can do the login in the system with the expired password, and the system will allow the attacker to change the password.
>
> I think the correct approach here is to lock down the account when the password expires. Therefore the user should change the password before the current password expires.
> If the user does not change the password in time, it must request the Administrator to enable the account with a new password.

Yes, I agree.

> > 1. How the user gets notified about it
> > When a user tries to log in, he gets notified about the expired password on the login page. He'll be asked to set a new password which will also update the *lastPasswordUpdate* field to the current time. It would be great if the implementation of https://tracker.ceph.com/issues/24655 can used here as well.
> >
> > Any further ideas/comments?
>
> I think we should issue a warning upon login a few (configurable) days before the password expires so that the user has the time to change the password.

Good idea!

> Also, this feature can only be implemented after https://tracker.ceph.com/issues/40248 is resolved, otherwise users will not be able to change their passwords.

#6 - 07/09/2019 03:29 PM - Lenz Grimmer

Thank you for this proposal and sorry for chiming in late here. The outcome of the conversation looks good to me (thanks to everyone who contributed), I'd like to point out one thing:

Tatjana Dehler wrote:

> 1. How the user gets notified about it
>   When a user tries to log in, he gets notified about the expired password on the login page. He'll be asked to set a new password which will also update the *lastPasswordUpdate* field to the current time. It would be great if the implementation of https://tracker.ceph.com/issues/24655 can used here as well.

Due to security reasons, the dashboard must not reveal any specific details about why the user could not log in on the login page itself (at least not on the login page, adding a more detailed reason to the mgr log would be fine). An attacker should not get any hint about **why** the login failed, e.g. if the username or password were incorrect, or if the password expired. This would give them a clue on how to further proceed. For example: an error like "wrong password" would reveal to me, that the username was correct. Similarly, "password expired" would also reveal to me that the user account exists (and I could use social engineering to convince the admin to reset the password). As long as the password is still valid, it's of course fine to remind the user that their password is about to expire and should be renewed

Should we make it configurable how many days in advance this warning appears?

#7 - 07/10/2019 06:27 AM - Tatjana Dehler

Lenz Grimmer wrote:

> Thank you for this proposal and sorry for chiming in late here. The outcome of the conversation looks good to me (thanks to everyone who contributed), I'd like to point out one thing:
>
> Tatjana Dehler wrote:
>
> > 1. How the user gets notified about it
> >   When a user tries to log in, he gets notified about the expired password on the login page. He'll be asked to set a new password which will also update the *lastPasswordUpdate* field to the current time. It would be great if the implementation of https://tracker.ceph.com/issues/24655 can used here as well.
>
> Due to security reasons, the dashboard must not reveal any specific details about why the user could not log in on the login page itself (at least

> not on the login page, adding a more detailed reason to the mgr log would be fine). An attacker should not get any hint about **why** the login failed, e.g. if the username or password were incorrect, or if the password expired. This would give them a clue on how to further proceed. For example: an error like "wrong password" would reveal to me, that the username was correct. Similarly, "password expired" would also reveal to me that the user account exists (and I could use social engineering to convince the admin to reset the password). As long as the password is still valid, it's of course fine to remind the user that their password is about to expire and should be renewed

Yes, makes sense to me.

> Should we make it configurable how many days in advance this warning appears?

Yes, that shouldn't be a problem.

#8 - 07/12/2019 03:50 PM - Lenz Grimmer

- Tags set to security

- Target version set to v15.0.0

#9 - 07/12/2019 03:52 PM - Lenz Grimmer

- Related to Feature #40248: mgr/dashboard: As a user, I want to change my password added

#10 - 07/12/2019 03:55 PM - Lenz Grimmer

- Related to Feature #25229: mgr/dashboard: Provide user enable/disable capability added

#11 - 07/12/2019 03:55 PM - Lenz Grimmer

- Related to Feature #24655: mgr/dashboard: Enforce password change upon first login added

#12 - 07/12/2019 03:55 PM - Lenz Grimmer

- Related to Feature #25232: mgr/dashboard: Support minimum password complexity rules added

#13 - 07/12/2019 03:57 PM - Lenz Grimmer

- Related to Feature #39999: mgr/dashboard: Prevent brute-force/dictionary attacks against existing local user accounts added

#14 - 07/29/2019 01:29 PM - Ricardo Marques

- Related to Fix #40328: mgr/dashboard: Permanent notifications instead of repeated notifications added

#15 - 08/16/2019 11:16 AM - Tatjana Dehler

- Assignee deleted (Tatjana Dehler)

#16 - 09/11/2019 01:28 PM - Tatjana Dehler

- File Screenshot_2019-09-11_15-23-50.png added

#17 - 09/16/2019 07:53 AM - Tatjana Dehler

- Assignee set to Tatjana Dehler

#18 - 10/16/2019 08:57 AM - Tatjana Dehler

- Status changed from New to In Progress

- Pull request ID set to 30939

#19 - 10/18/2019 07:42 AM - Tatjana Dehler

- Description updated

#20 - 10/30/2019 04:11 PM - Tatjana Dehler

> Further questions:
>
> - admin password expiry: Should it be possible to set an expiry date for the admin password as well? Or only if there is at least another admin account? If it should not be possible to set expiry date prevent the user from doing so.

Passwords of admin accounts should never expire in my opinion.

> - disabled users password expiry: Should it be possible to set/have an expiry date for disabled users?

It should not be possible to set an expiry date for disabled users in my opinion. While disabling a user account the expiry date field should be cleared if set.

> - 'ac_user_create_cmd' requires timestamp as 'pwd_expiry_date': The function (ac_user_create_cmd) to create a user on the command line requires a timestamp as 'pwd_expiry_date' at the moment. Do we want to keep it or change the behavior here?

?

> - recalculate password expiry date: issue https://tracker.ceph.com/issues/40329 introduces a default expiry span (USER_PWD_DEFAULT_EXPIRY_SPAN) for the user passwords and adds a password expiry date field (pwd_expiry_date) to the User class. If the administrator edits the USER_PWD_DEFAULT_EXPIRY_SPAN variable the password expiry dates need to be re-calculated.

?

> - update password expiry date (which is set manually): If the 'USER_PWD_DEFAULT_EXPIRY_SPAN' is set and the user changes the password, it's easy to update the expiry date to the next date. But what happens if 'USER_PWD_DEFAULT_EXPIRY_SPAN' is not set and the password expiry date was entered manually?

I would assume the password expiry date will be cleared.

#21 - 12/16/2019 09:50 AM - Lenz Grimmer

- Status changed from In Progress to Resolved

#22 - 12/27/2019 04:26 AM - Kefu Chai

- Related to Bug #43431: mgr/dashboard: test_create_with_default_expiration_date (tasks.mgr.dashboard.test_user.UserTest) added

#23 - 08/05/2020 02:37 PM - Ernesto Puerta

- Status changed from Resolved to Pending Backport

- Backport set to nautilus

#24 - 08/05/2020 02:38 PM - Ernesto Puerta

- Copied to Backport #46837: nautilus: mgr/dashboard: user management improvements (password change, password complexity, ...) added

#25 - 09/21/2020 02:50 PM - Ernesto Puerta

- Status changed from Pending Backport to Closed

- Backport deleted (nautilus)

For clean/safe backport it requires more than 11 additionall PRs

Closing.

#26 - 10/06/2020 10:45 AM - Ernesto Puerta

- Parent task set to #47765

Files

Screenshot_2019-09-11_15-23-50.png	60 KB	09/11/2019	Tatjana Dehler