

✓ CentralAuth expiring global groups do not expire (CVE-2022-28205)

Actions

✓ Closed, Resolved

Public

SECURITY

Assigned To

Zabe

Authored By

Urbanecm

2022-02-21 21:08:03 (UTC+0)

Tags

Security-Team (Watching)

Security


MediaWiki-extensions-CentralAuth (Incoming)


SecTeam-Processed (Completed)


MW-1.38-notes (1.38.0-wmf.24; 2022-02-28)


Vuln-MissingAuthz (Tracked)

Referenced Files

 **F34963071: 0001-SECURITY-Ignore-cached-CentralAuthUser-entries-with-.patch**
2022-02-23 13:44:55 (UTC+0)

 **F34963049: 0001-SECURITY-Ignore-cached-CentralAuthUser-entries-with-.patch**
2022-02-23 13:24:43 (UTC+0)

 **F34961333: 0001-SECURITY-Fix-ttl-for-groups-expiring-in-the-future.patch**
2022-02-22 00:10:50 (UTC+0)

 **F34961190: image.png**
2022-02-21 21:08:03 (UTC+0)

Subscribers

Aklapper

Legoktm

sbassett

Superpes15

taavi

Tks4Fish

Urbanecm

Zabe

Description

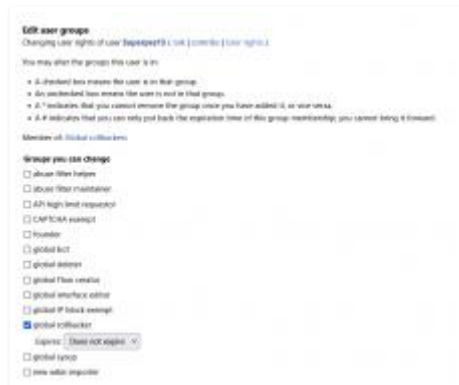
Hello,

earlier today, I was alerted by [@Superpes15](#) that his temporarily granted global group (Abuse filter helpers) that is supposed to expire on 12:39, 20 February 2022 (slightly over 24 hours as of writing) is still shown at <https://meta.wikimedia.org/wiki/Special:CentralAuth?target=Superpes15>.

Worse, MW apparently thinks Superpes15 is still in the group:

```
[urbanecm@mwmain1002 ~]$ mwscrip shell.php enwiki
Psy Shell v0.11.1 (PHP 7.2.34-18+0~20210223.60+debian10~1.gbp21322+wmf5 - cli) by Justin Hileman
>>> User::newFromName('Superpes15')->isAllowed('abusefilter-view-private')
=> true
>>> User::newFromName('Superpes15')->isAllowed('abusefilter-log-private')
=> true
>>>
```

<https://meta.wikimedia.org/wiki/Special:GlobalUserRights/Superpes15> only shows the "Global rollbacker" group, as it should:



I'm intentionally not trying to fix the group membership for Superpes15's account, as Superpes15 is trusted enough to have the rights for a couple of more days/hours and it might make it easier for us to debug the issue.

CC @Majavah, who created the feature.

Details



Risk Rating

Low

Author Affiliation

Project

Subject

-  [mediawiki/extensions/CentralAuth](#) [SECURITY: Ignore cached CentralAuthUser entries with expired groups](#)
-  [mediawiki/extensions/CentralAuth](#) [SECURITY: Fix ttl for groups expiring in the future](#)

[Customize query in Gerrit](#)

Related Objects






Mentions

Mentioned In

~~T297839: Write and send supplementary release announcement for extensions and skins with security patches (1.35.6/1.36.4/1.37.2)~~

Mentioned Here

~~T297839: Write and send supplementary release announcement for extensions and skins with security patches (1.35.6/1.36.4/1.37.2)~~

-  **Urbanecm** created this task. 2022-02-21 21:08:03 (UTC+0)
-   Restricted Application added a subscriber: **Aklapper**. · [View Herald Transcript](#) 2022-02-21 21:08:04 (UTC+0)
-  **Urbanecm** added a project: **MediaWiki-extensions-CentralAuth**. 2022-02-21 21:08:21 (UTC+0)
-  **Urbanecm** updated the task description. ([Show Details](#))

→ **taavi** triaged this task as *High* priority. 2022-02-21 21:18:24 (UTC+0)

 **taavi** added a subscriber: **Zabe**.

I suspect this is a caching issue. CentralAuthUser is supposed to ensure any user data cache TTLs are maxed at the closest user group expiry (see `CentralAuthUser::loadFromCache` and `CentralAuthUser::getClosestGlobalUserGroupExpiry`).

While debugging this I found some very strange behaviour:

```
taavi@mwmaint1002 ~ $ mwscrip shell.php metawiki
Psy Shell v0.11.1 (PHP 7.2.34-18+0~20210223.60+debian10~1.gbpb21322+wmf5 - cli) by Justin Hileman
>>> User::newFromName('Superpes15')->isAllowed('abusefilter-view-private')
=> true
>>> ^D
Exit: Ctrl+D
taavi@mwmaint1002 ~ $ mwscrip shell.php metawiki
Psy Shell v0.11.1 (PHP 7.2.34-18+0~20210223.60+debian10~1.gbpb21322+wmf5 - cli) by Justin Hileman
>>> \Wikimedia\TestingAccessWrapper::newFromObject(
\MediaWiki\Extension\CentralAuth\User\CentralAuthUser::getInstanceByName( 'Superpes15' ) )->
getClosestGlobalUserGroupExpiry()
=> null
>>> User::newFromName('Superpes15')->isAllowed('abusefilter-view-private')
```

```
=> false
>>> ^D
Exit: Ctrl+D
```

🗨 **taavi** added a comment. 2022-02-21 21:26:19 (UTC+0)

```
>>> $cache->get( $user->getCacheKey( $cache ) )['mGroupExpirations']
=> [
    "abusefilter-helper" => "20220220113921",
    "global-rollbacker" => null,
]
```

Something's going very wrong here, since even the original TTL is set to a day (and the logic I mentioned above can only shorten it), as if something was just ignoring it.

Do we need to patch CentralAuth to ignore any cache entries that contain user groups that expire in the past?

🗨 **Urbanecm** added a comment. 2022-02-21 21:29:31 (UTC+0)

I confirm that weird behavior referenced at **T302248#7726529**. `CentralAuthUser::getInstance($u) ->getGlobalRights()` "sometimes" return the wrong set of permissions for Superpes.
`getClosestGlobalUserGroupExpiry()` returns non-null.

```
[urbanecm@mwmaint1002 ~]$ mwscrip shell.php dewiki
Psy Shell v0.11.1 (PHP 7.2.34-18+0~20210223.60+debian10~1.gbp21322+wmf5 - cli) by Justin Hileman
>>> $u = User::newFromName('Superpes15')
=> User {#3306
    +mId: null,
    +mName: "Superpes15",
    +mActorId: null,
    +mRealName: null,
    +mEmail: null,
    +mTouched: null,
    +mEmailAuthenticated: null,
    +mFrom: "name",
    +mBlockedby: -1,
    +mHideName: null,
    +mBlock: null,
}
>>> $u->isAllowed('abusefilter-view-private')
=> true
>>> use MediaWiki\Extension\CentralAuth\User\CentralAuthUser
>>> $cu = CentralAuthUser::getInstance($u)
=> MediaWiki\Extension\CentralAuth\User\CentralAuthUser {#3365
    +mStateDirty: false,
    +mHomeWiki: "itwiki",
}
>>> $cu->getGlobalRights()
=> [
    "abusefilter-log",
    "abusefilter-log-detail",
    "abusefilter-log-private",
    "abusefilter-view",
    "abusefilter-view-private",
    "spamblacklistlog",
```

```
"abusefilter-log-detail",
"autoconfirmed",
"autopatrol",
"autoreviewrestore",
"editsemiprotected",
"markbotedits",
"move",
"movestable",
"noratelimit",
"patrolmarks",
"rollback",
"skipcaptcha",
"suppressredirect",
]
>>> sudo $cu->getClosestGlobalUserGroupExpiry()
=> 1645357161
>>>
```

re-running the same snippet in a new shell.php session works as expected (w/o the additional rights being allowed).
So...apparently, cache gets updated, but *after* it is consumed?

 **Tks4Fish** added a subscriber: **Tks4Fish**. 2022-02-21 22:14:18 (UTC+0)

 **Urbanecm** added a comment. 2022-02-21 22:15:08 (UTC+0) 

In **T302248#7726551**, @Majavah wrote:

```
>>> $cache->get( $user->getCacheKey( $cache ) )['mGroupExpirations']
=> [
    "abusefilter-helper" => "20220220113921",
    "global-rollbacker" => null,
]
```

Something's going very wrong here, since even the original TTL is set to a day (and the logic I mentioned above can only shorten it), as if something was just ignoring it.

CentralAuthUser::loadFromCache uses the following code to set the TTL:

```
$closestGugExpiry = $this->getClosestGlobalUserGroupExpiry();
if ( $closestGugExpiry ) {
    $ttl = min( time() - $closestGugExpiry, $ttl );
}
```

AFAICS `time() - $closestGugExpiry` is going to be negative when `getClosestGlobalUserGroupExpiry` returns a timestamp that's in the future. If I understand the path correctly, it ends up setting a negative TTL since `$closestGugExpiry` is expected to be in the future (and I'm not sure how are negative TTLs treated by core).

 **Zabe** added a comment. Edited · 2022-02-22 00:10:50 (UTC+0) 

In **T302248#7726667**, @Urbanecm wrote:

In **T302248#7726551**, @Majavah wrote:

```
>>> $cache->get( $user->getCacheKey( $cache ) )['mGroupExpirations']
=> [
    "abusefilter-helper" => "20220220113921",
    "global-rollbacker" => null,
]
```

Something's going very wrong here, since even the original TTL is set to a day (and the logic I mentioned above can only shorten it), as if something was just ignoring it.

CentralAuthUser::loadFromCache uses the following code to set the TTL:

```
$closestGugExpiry = $this->getClosestGlobalUserGroupExpiry();
if ( $closestGugExpiry ) {
    $ttl = min( time() - $closestGugExpiry, $ttl );
}
```

AFAICS `time() - $closestGugExpiry` is going to be negative when `getClosestGlobalUserGroupExpiry` returns a timestamp that's in the future. If I understand the path correctly, it ends up setting a negative TTL since `$closestGugExpiry` is expected to be in the future (and I'm not sure how are negative TTLs treated by core).

Yeah, that definitely doesn't seem correct. It also is the other way around in core:

<https://gerrit.wikimedia.org/r/plugins/gitiles/mediawiki/core/+/aeaa3a582e1df47d1aa9cf2febf3c20d0ba1ca9f/includes/user/User.php#545>

Not sure if that is the only issue, but it definitely seems worth fixing.

Proposed patch:



0001-SECURITY-Fix-ttl-for-groups-expiring-in-the-future.patch 1 KB

Download

 **Zabe** added a project: **Patch-For-Review**. 2022-02-22 00:20:37 (UTC+0)

 **taavi** added a comment. 2022-02-22 14:15:45 (UTC+0)

In **T302248#7726840**, @Zabe wrote:



0001-SECURITY-Fix-ttl-for-groups-expiring-in-the-future.patch 1 KB

Download

Deployed.

 **Zabe** removed a project: **Patch-For-Review**. 2022-02-22 14:18:37 (UTC+0)

 **sbassett** mentioned this in ~~T297839: Write and send supplementary release announcement for extensions and skins with security patches (1.35.6/1.36.4/1.37.2)~~. 2022-02-22 16:41:58 (UTC+0)

 **sbassett** added a subscriber: **sbassett**. 2022-02-22 16:46:34 (UTC+0)

In ~~T302248#7728138~~, @Majavah wrote:

Deployed.

Thanks. [SAL](#). Also tracking at T276237 and **T297839**.

 **sbassett** moved this task from **Incoming** to **Watching** on the **Security-Team** board. 2022-02-22 16:46:56 (UTC+0)

 **sbassett** added a project: **SecTeam-Processed**.

 **taavi** added a comment. 2022-02-22 20:28:40 (UTC+0)

This problem seems to have fixed itself in production in the meantime:

```
>>> $users = CentralAuthServices::getDatabaseManager()->getCentralDB( DB_REPLICA )-
>selectFieldValues( 'global_user_groups', 'gug_user', [ 'gug_expiry < now()', 'gug_group' =>
'abusefilter-helper' ] );
=> [
    "2645",
    "6677077",
    "44717363",
    "55412109",
    "56429570",
    "59866209",
]
>>> foreach ( $users as $id ) { var_dump( in_array( 'abusefilter-view-private',
\MediaWiki\Extension\CentralAuth\User\CentralAuthUser::newFromId( (int)$id )->getGlobalRights() ) );
}
bool(false)
bool(false)
bool(false)
bool(false)
bool(false)
bool(false)
bool(false)
>>> User::newFromName( 'Superpes15' )->isAllowed( 'abusefilter-view-private' )
=> false
```

With the above fix deployed, is there anything left to do here?

In ~~T302248#7728975~~, @**sbassett** wrote:

*Thanks. [SAL](#). Also tracking at T276237 and **T297839**.*

Note that this feature didn't make it into a release branch yet.

 **Urbanecm** added a comment. 2022-02-22 20:54:47 (UTC+0)

In **T302248#7730003**, @Majavah wrote:

This problem seems to have fixed itself in production in the meantime:

```
>>> $users = CentralAuthServices::getDatabaseManager()->getCentralDB( DB_REPLICA )-
>selectFieldValues( 'global_user_groups', 'gug_user', [ 'gug_expiry < now()', 'gug_group' =>
'abusefilter-helper' ] );
=> [
    "2645",
    "6677077",
    "44717363",
    "55412109",
    "56429570",
    "59866209",
]
>>> foreach ( $users as $id ) { var_dump( in_array( 'abusefilter-view-private',
\MediaWiki\Extension\CentralAuth\User\CentralAuthUser::newFromId( (int)$id )->getGlobalRights()
) ); }
bool(false)
bool(false)
bool(false)
bool(false)
bool(false)
bool(false)
bool(false)
>>> User::newFromName( 'Superpes15' )->isAllowed( 'abusefilter-view-private' )
=> false
```

With the above fix deployed, is there anything left to do here?

I think ensure it was the cause. We should probably discuss the hardening you described at **T302248#7726551** (I'm not sure about it; on one hand, it decreases the probability of this happening again, but it also makes it harder to fix the actual error in the code).

 **sbassett** added a comment. 2022-02-22 20:57:46 (UTC+0)

In **T302248#7730106**, @Urbanecm wrote:

With the above fix deployed, is there anything left to do here?

*I think ensure it was the cause. We should probably discuss the hardening you described at **T302248#7726551** (I'm not sure about it; on one hand, it decreases the probability of this happening again, but it also makes it harder to fix the actual error in the code).*

Yes, that. But assuming things have been fixed well enough for now, there's no reason why this task couldn't be made public (I'm not seeing any obvious PII or sensitive info) and the backports pushed through gerrit so that the patch lands on a proper release branch next week, or whenever. Otherwise that will definitely happen once the next supplemental security release is sent out towards the end of March 2022.

 **Legoktm** added a subscriber: **Legoktm**. 2022-02-23 01:52:53 (UTC+0)

In **T302248#7726551**, @Majavah wrote:


```
>>> $cache->get( $user->getCacheKey( $cache ) )['mGroupExpirations']
=> [
    "abusefilter-helper" => "20220220113921",
    "global-rollbacker" => null,
]
```

Something's going very wrong here, since even the original TTL is set to a day (and the logic I mentioned above can only shorten it), as if something was just ignoring it.

Do we need to patch CentralAuth to ignore any cache entries that contain user groups that expire in the past?

This seems like a good hardening measure to me, rather than relying on how well cache expiry works.

Urbanecm added a comment. 2022-02-23 12:18:56 (UTC+0)

In **T302248#7730117**, @sbassett wrote:

[...]

Yes, that. But assuming things have been fixed well enough for now, there's no reason why this task couldn't be made public (I'm not seeing any obvious PII or sensitive info) and the backports pushed through gerrit so that the patch lands on a proper release branch next week, or whenever. Otherwise that will definitely happen once the next supplemental security release is sent out towards the end of March 2022.

I added a temporary global group to my alt account (<https://meta.wikimedia.org/w/index.php?title=Special:Log&logid=46210369>). I plan to check tomorrow the group disappeared. We don't have exact ways to reproduce unfortunately, but it's at least something :).

taavi added a project: **Patch-For-Review**. 2022-02-23 13:24:43 (UTC+0)

In **T302248#7730649**, @Legoktm wrote:

In **T302248#7726551**, @Majavah wrote:

Do we need to patch CentralAuth to ignore any cache entries that contain user groups that expire in the past?

This seems like a good hardening measure to me, rather than relying on how well cache expiry works.

Proposed patch:



0001-SECURITY-Ignore-cached-CentralAuthUser-entries-with-.patch 2 KB

Download

Zabe added a comment. Edited · 2022-02-23 13:40:14 (UTC+0)

In **T302248#7731921**, @Majavah wrote:

Proposed patch:



0001-SECURITY-Ignore-cached-CentralAuthUser-entries-with-.patch 2 KB

Download

-1, `getClosestGlobalUserGroupExpiry` returns null for users without temporary groups, resulting in their groups being reloaded in every request.

taavi added a comment. 2022-02-23 13:44:55 (UTC+0)

In **T302248#7731942**, **@Zabe** wrote:

-1, `getClosestGlobalUserGroupExpiry` returns null for users without temporary groups, resulting in their groups being reloaded in every request.

Thanks, fixed.



0001-SECURITY-Ignore-cached-CentralAuthUser-entries-with-.patch 2 KB

Download

Zabe added a comment. Edited · 2022-02-23 13:55:18 (UTC+0)

In **T302248#7731951**, **@Majavah** wrote:

In **T302248#7731942**, **@Zabe** wrote:

-1, `getClosestGlobalUserGroupExpiry` returns null for users without temporary groups, resulting in their groups being reloaded in every request.

Thanks, fixed.



0001-SECURITY-Ignore-cached-CentralAuthUser-entries-with-.patch 2 KB

Download

+2

There is a typo in the @param statement, but that doesn't matter too much, since it is a sec patch.

taavi added a comment. 2022-02-23 14:26:17 (UTC+0)

In **T302248#7731979**, **@Zabe** wrote:

In **T302248#7731951**, **@Majavah** wrote:

Thanks, fixed.



0001-SECURITY-Ignore-cached-CentralAuthUser-entries-with-.patch 2 KB

Download

+2

Deployed too. I'll fix that typo when pushing these to Gerrit.

In **T302248#7730117**, @sbassett wrote:

Yes, that. But assuming things have been fixed well enough for now, there's no reason why this task couldn't be made public (I'm not seeing any obvious PII or sensitive info) and the backports pushed through gerrit so that the patch lands on a proper release branch next week, or whenever. Otherwise that will definitely happen once the next supplemental security release is sent out towards the end of March 2022.

Sounds good. The hardening patch also added log message that should warn us if this is happening again (at least the only known case was a caching issue, I really hope there aren't any other logic bugs here). I don't think there's much exploitable if this is made public, as an attacker would need to temporarily get added to global group by a steward in the first place.

Zabe removed a project: **Patch-For-Review**. 2022-02-23 14:53:48 (UTC+0)

sbassett changed the visibility from "**Custom Policy**" to "Public (No Login Required)".

2022-02-23 21:17:40 (UTC+0)

sbassett changed the edit policy from "**Custom Policy**" to "All Users".

sbassett changed Risk Rating from N/A to Low.

In **T302248#7732141**, @Majavah wrote:

Deployed too. I'll fix that typo when pushing these to Gerrit.

Thanks.

Sounds good. The hardening patch also added log message that should warn us if this is happening again (at least the only known case was a caching issue, I really hope there aren't any other logic bugs here). I don't think there's much exploitable if this is made public, as an attacker would need to temporarily get added to global group by a steward in the first place.

This is now public, so we can get the backports going in gerrit.

gerritbot added a comment. 2022-02-23 21:20:59 (UTC+0)

Change 765335 had a related patch set uploaded (by SBassett; author: Zabe):

[mediawiki/extensions/CentralAuth@master] SECURITY: Fix ttl for groups expiring in the future


<https://gerrit.wikimedia.org/r/765335>

 **gerritbot** added a project: **Patch-For-Review**. 2022-02-23 21:20:59 (UTC+0) ▼

Change 765336 had a related patch set uploaded (by Zabe; author: Majavah):

[mediawiki/extensions/CentralAuth@master] SECURITY: Ignore cached CentralAuthUser entries with expired groups

<https://gerrit.wikimedia.org/r/765336>

 **gerritbot** added a comment. 2022-02-23 21:28:10 (UTC+0) ▼

Change 765335 **merged** by jenkins-bot:

[mediawiki/extensions/CentralAuth@master] SECURITY: Fix ttl for groups expiring in the future

<https://gerrit.wikimedia.org/r/765335>


 **gerritbot** added a comment. 2022-02-23 21:46:14 (UTC+0) ▼

Change 765336 **merged** by jenkins-bot:

[mediawiki/extensions/CentralAuth@master] SECURITY: Ignore cached CentralAuthUser entries with expired groups

<https://gerrit.wikimedia.org/r/765336>


 **ReleaseTaggerBot** added a project: ~~MW-1.38-notes (1.38.0-wmf.24, 2022-02-28)~~. 2022-02-23 22:00:17 (UTC+0)

 **Zabe** closed this task as *Resolved*. 2022-02-24 14:47:26 (UTC+0)

 **Zabe** claimed this task.

 **Zabe** removed a project: **Patch-For-Review**.

 **sbassett** added a project: **Vuln-MissingAuthz**. 2022-03-29 01:32:38 (UTC+0)

 **sbassett** renamed this task from *CentralAuth expiring global groups do not expire* to *CentralAuth expiring global groups do not expire (CVE-2022-28205)*. 2022-03-30 19:20:48 (UTC+0)