

[security:high, CVE-2021-35472] session cache corruption can lead to authorization bypass or spoofing

Concerned version

Version: %2.0.0 to %2.0.11

Platform: Nginx/uWSGI

Summary

- Enable Impersonation plugin
- Enable REST Session server
- Disable CSRF tokens
- Start a terminal and execute : for i in {1..1000}; do curl -X POST -H 'Accept:application/json' -d user=msmith --data-urlencode password='msmith' http://auth.example.com:19876;done
- make reload
- Login dwho/dwho/dwho and hit FS to refresh Portal
- Alternatively authenticated as 'dwho' or 'msmith'

Backends used

PG vokoscreen-2021-06-08 22-12-00

Possible fixes

Seems issue is linked to handler internal cache.

Login with 'dwho' / 'dwho'

Enable Impersonation plugin -> make reload_web_server

Start bash loop, hit F5 and session switches to 'msmith'

Stop bash loop and session is back to 'dwho' after 10/15 seconds.:

Edited 1 year ago by [Yadd](#)

⬆️ Drag your designs here or [click to upload](#)

Tasks 0


No tasks are currently assigned. Use tasks to break down this issue into smaller parts.

Linked items 0

Link issues together to show that they're related. [Learn more](#)

Activity

- ① Christophe Maudoux changed milestone to [%2.0.12](#) 1 year ago
- 🔗 Christophe Maudoux added [Bug](#), [Portal](#), [Security](#), labels 1 year ago
- 🔗 Christophe Maudoux changed the description 1 year ago
- 🔗 Christophe Maudoux changed the description 1 year ago

 **Maxime Besson** @maxbex · 1 year ago

The issue occurs when Impersonation.pm calls

```
$self->p->updateSession( $req, $spoorSession );
```

which updates the handler cache






```
$req->(sessionInfo)->{$_} = $self->HANDLER->data->{$_} =  
$req->(sessionInfo)->{$_} = $infos->{$_};
```

but only partially. `_session_id` keeps the previous value (`dwho` 's session), but the rest of the attributes are updated to `msmith`. That is why `dwho` ends up with `msmith` 's attributes.

No more time for today, but we need to find out why the session id is not updated in the handler cache at the end of the login process. There might be more serious security issues as a consequence of that.

Edited by **Maxime Besson** · 1 year ago

 **Christophe Maudoux** @maudoux · 1 year ago
Hi @mauber
Many thx for your help and investigations...
Cheers

Maintain

Alright, as I suspected, the way the handler cache is updated in the portal can lead to real security issues in production.

Here is a proof of concept with:

- a CAS issuer relying on `authnLevel`
- a 2FA login (2FA does an `updateSession` to raise the `authnLevel`)

0:00 / 0:38

[@volkoscreeen-2021-06-16 21:49:41](#)

On the right: an attacker (dwho) with a valid session at `authnLevel 2` fails to obtain a valid CAS ticket because his `authnLevel` is too low. On the left: a user (msmith) logs in with `authnLevel=5`

Just after validating the OTP code, `dwho's` attempt succeeds, and `dwho` gets a valid CAS ticket for the service (for `dwho's` session, thankfully, but this is still a bypass)

In this test, `Impersonation` was not enabled.

Any plugin that uses `updateSession` may lead to those mixups, but so far, the most risky ones are:

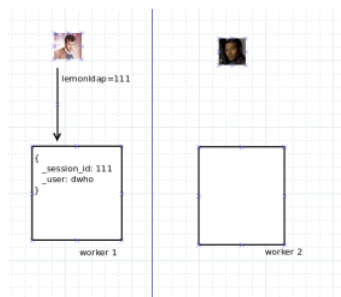
- `Impersonation`: updates the whole session
- `SecondFactor`: updates `authnLevel`, groups and macros

- [Maxime Besson](#) changed title from **Resquests are mixed up if impersonation plugin is enabled** to **session cache corruption can lead to authorization bypass or spoofing** 1 year ago

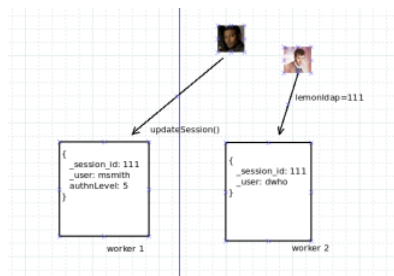
 [Maxime Besson](#) @maxbes · 1 year ago Maintainer

Here is a simplified explanation

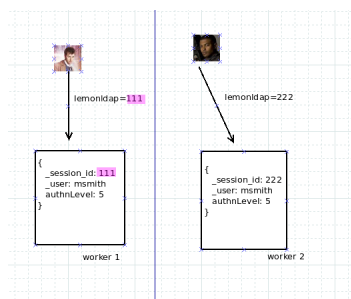
dwho is already logged in, and filling the handler cache with his session info



msmith is logging in, and in the process, calls updateSession with his new authnLevel. This call updates the handler cache, but only _user and authnLevel, not _session_id



next time dwho hits the same worker, his session ID matches _session_id in the cache, and because of that retrieveSession fills the request's session info with msmith's authnLevel



I'm not sure how to fix this:

- modify updateSession to no longer modify the handler cache? But are there other functions that modify the handler cache?
- save a consistent handler cache at the end of each portal request? (we already have importHandlerData at the beginning of processing, we could add exportHandlerData at the end)
- remove the handler cache completely from the portal? We probably don't really need it

@quimard, what do you think?

Edited by Maxime Besson 1 year ago



Maxime Besson @maxbes · 1 year ago

Maintainer

SAML issuer is also affected, with a risk of identity spoofing, because it builds the response assertion directly from the current request's session info (which comes from corrupted cache)

Variable	Value
<code>\$_COOKIE['mellon-cookie']</code>	<code>a7e0d07c1157714f3c608550c409e4</code>
<code>\$_SERVER['MELLON_NAME_ID']</code>	<code>_OCF122798248F613593FF094D96508C0</code>
<code>\$_SERVER['MELLON_NAME_ID_0']</code>	<code>_OCF122798248F613593FF094D96508C0</code>
<code>\$_SERVER['MELLON_whatToTrace']</code>	<code>msmith</code>
<code>\$_SERVER['MELLON_whatToTrace_0']</code>	<code>msmith</code>
<code>\$_SERVER['MELLON_user']</code>	<code>dwho</code>
<code>\$_SERVER['MELLON_user_0']</code>	<code>dwho</code>

Edited by Maxime Besson 1 year ago



Yadd @quimard · 1 year ago

Owner

Hi @maxbes,

could you reserve a CVE? Also the fix commits should be separated from any other change: it will help me to fix versions 2.0.2 and 2.0.11 (Debian versions).

Cheers, Yadd



Maxime Besson changed title from **session cache corruption can lead to authorization bypass or spoofing** to **[security-high] session cache corruption can lead to authorization bypass or spoofing** 1 year ago



Maxime Besson @maxbes · 1 year ago

Maintainer

CVSSv3 score: 7.7 (AV:N/AC:H/PR:L/UI:R/S:C/CH/I/H/AN)

I want to wait a little before getting a CVE, for now only people using 2FA (and impersonation but that's not surprising) seem to be affected, but there might be other triggers. I want the CVE description to be as complete as possible.

We also need to find a solution.

Edited by Maxime Besson 1 year ago



Yadd @quimard · 1 year ago

Owner

First we should write a test to reproduce this issue



Maxime Besson @maxbes · 1 year ago

Maintainer

This unit test demonstrates the issue in the 2FA system

[@_test-25391](#)



Yadd @quimard · 1 year ago

Owner

Hi,

I think this fix is enough: clear internal cache if no id :

```
--- a/lemonldap-ng-handler/11b/lemonldap-ng/Handler/Main/Run.pm
+++ b/lemonldap-ng-handler/11b/lemonldap-ng/Handler/Main/Run.pm
```

```
@@ -139,7 +139,9 @@ sub run {
    }

    # Try to recover cookie and user session
    - if ( $sid = $class->fetchId($req)
    + $sid = $class->fetchId($req);
    + $class->data( {} ) unless($sid);
    + if ( $sid
      and $session = $class->retrieveSession( $req, $sid ) )
    {
```

 [Yadd @quimard](#) · 1 year ago

Owner


This keep wanted behavior: internal cache contains last session.

 [Yadd @quimard](#) · 1 year ago

Owner

[@maxbes](#): is it OK for you ?

Please [register](#) or [sign in](#) to reply

 [Maxime Besson](#) mentioned in commit [2de2cbf4](#) 1 year ago



[Maxime Besson @maxbes](#) · 1 year ago

Maintainer

Looks good. As an extra precaution it might also be good to change updateSession so that it will only update the handler cache if the target session matches the ID in the cache:

```
$req->(sessionInfo->){$_} = $infos->{$_};
if ($sid eq $se1F->HANDLER->data->{ session_id }) {
    $se1F->HANDLER->data->{$_} = $infos->{$_};
}
```

but doing this breaks some ContextSwitching unit tests

Funlly, your fix breaks a unit test which relied on the vulnerability to work (the test incorrectly used an old session ID after a session upgrade!). I fixed the unit test.

I couldn't find another exploitation condition than these two so far:

- Impersonation/ContextSwitching plugins
- Secondfactor with authLevel set

A user on the mailing list has reported an issue that looks very much like this one. But I'm waiting for their feedback to be sure they are using 2FA. If they aren't, it means I'm missing another possible vector. However clearing the cache for unauthenticated requests this should be enough to mitigate.

 [Yadd @quimard](#) · 1 year ago

Owner


[@maudoux](#): could you help here ?

 [Yadd @quimard](#) · 1 year ago

Owner

Failing tests:

- t/68-ContextSwitching-with-2F-allowed.t
- t/68-ContextSwitching-with-Impersonation.t

 [Christophe Maudoux @maudoux](#) · 1 year ago

Author

Maintainer

I will take a look asap...

 [Christophe Maudoux @maudoux](#) · 1 year ago

Author

Maintainer

Hi [@quimard](#)

Seems all unit tests are OK...

 [Yadd @quimard](#) · 1 year ago

Owner

[@maudoux](#): not with the proposed patch

 [Christophe Maudoux @maudoux](#) · 1 year ago

Author

Maintainer

OK! Good point 🙄

Need to dig more...

Edited by [Christophe Maudoux](#) 1 year ago

 [Yadd @quimard](#) · 1 year ago

Owner

[@maudoux](#): I just committed [@maxbes](#) patch, then you can see which tests fail

 [Christophe Maudoux @maudoux](#) · 1 year ago

Author

Maintainer

Ok will be done this we...

Please [register](#) or [sign in](#) to reply



[Yadd @quimard](#) · 1 year ago


Owner

[@maxbes](#): so can I commit now ?

 [Yadd @quimard](#) · 1 year ago

Owner

I'll update updateSession after, it requires more work

 [Yadd @quimard](#) · 1 year ago


Owner

Test renamed to t/91-handler-cache-cleaned.t

 [Maxime Besson @maxbes](#) · 1 year ago

Maintainer

Yes you can commit, if I don't have any feedback next week I'll do a CVE request with the info we currently have


 [Yadd @quimard](#) · 1 year ago

Owner

High impact but low probability

Please [register](#) or [sign in](#) to reply


 [Yadd](#) mentioned in commit [b6a1f946](#) 1 year ago

 [Maxime Besson](#) changed title from [\[security:high\] session cache corruption can lead to authorization bypass or spoofing](#) to [\[security:high, CVE-2021-35472\] session cache corruption can lead to authorization bypass or spoofing](#) 1 year ago

 [Maxime Besson](#) changed the description 1 year ago

 [Clément OUDOT](#) mentioned in issue [#2112](#) 1 year ago

 [Clément OUDOT](#) assigned to [@quimard](#) 1 year ago

 [Maxime Besson](#) mentioned in commit [1d88c262](#) 1 year ago

 [Yadd](#) mentioned in commit [15954b98](#) 1 year ago



[Yadd @quimard](#) · 1 year ago

Owner

Debian updates are ready to push with all security fixes. Of course I'll wait for [%2.0.12](#) release.



[Yadd @quimard](#) · 1 year ago

Owner

Version 1.9.x has the same bug but is it a security issue ? Here is the patch:

```
--- a/lemonldap-ng-handler/lib/Lemonldap/NG/Handler/Main.pm
+++ b/lemonldap-ng-handler/lib/Lemonldap/NG/Handler/Main.pm
@@ -378,7 +378,9 @@
     my $sid;

     # Try to recover cookie and user session
-    if ( $sid = $class->fetchId
+    if ( $sid = $class->fetchId;
+    + $datas = {} unless $sid;
+    + if ( $sid
       and $class->retrieveSession($sid) )
     {
```



Maxime Besson @maxbes · 1 year ago

Maintainer

updateSession does not seem to update the handler cache in 1.9, but I'm not familiar enough with the code to be sure. Maybe the handler cache is updated in other places in 1.9?



Yadd @quimard · 1 year ago

Owner

@maxbes: you're right, portal and handler are totally separated



Yadd @quimard · 1 year ago

Owner

So this CVE is for ≥ 2.0.0



Yadd changed the description 1 year ago



Yadd closed 1 year ago



Yadd @quimard · 1 year ago

Owner

Reopened: waiting for @maudoux fix



Yadd reopened 1 year ago



Christophe Maudoux mentioned in commit 71ed63a0 1 year ago



Christophe Maudoux @maudoux · 1 year ago

Author

Maintainer

Done! 🎉



Christophe Maudoux closed 1 year ago



Clément OUDOT made the issue visible to everyone 1 year ago



Yadd @quimard · 1 year ago

Owner

Debian updates:

- Buster: 2.0.2+ds-7+deb10u6
- Bullseye: 2.0.11+ds-4

Please [register](#) or [sign in](#) to reply