

👁 Logins to MW with at least one SSO client extension allows masquerading as another user (CVE-2020-35623)

🔒 Closed, Resolved

🌐 Public

SECURITY

Actions

Assigned To

Sudozero

Authored By

Sudozero  
2020-09-21 20:57:56 (UTC+0)

Tags

👤 Security-Team (Our Part Is Done)

👤 Security

👤 MediaWiki-Authentication-and-authorization (Backlog)

👤 Vuln-Authn/Session (Tracked)

Referenced Files

None

Subscribers

Aklapper

Reedy

sbassett

Sudozero

Tokens

Description

I was able to successfully masquerade with another one of my user accounts using the [CASAAuth](#) extension. While that extension has been abandoned, other SSO client extensions may be at risk.

To log in as "foobar", one only need to create an account like: "\_foobar" or "foobar\_" with the SSO provider, and log in with that user name. As a result, one gains access to the "foobar" account. If one's account is "foo\_bar", then "foo\_bar" and "foo bar" also work.

By gaining access to an admin account, one could delete arbitrary pages or otherwise vandalize the site.

This appears to be due to the automatic normalization performed by MW code when creating user names ( `User::newFromName` ). The workaround I'm using for now is:

```
--- CASAAuth.php.orig 2020-09-21 16:50:52.757932797 -0400
+++ CASAAuth.php 2020-09-21 16:29:07.448092263 -0400
@@ -117,6 +117,17 @@
         return true;
     }

+    $collision_0 = "/^/";
+    $collision_1 = "/$/";
+    $collision_2 = "/_$/";
+    $collision_3 = "/ /";
+    if(preg_match($collision_0, $username) || preg_match($collision_1, $username) || preg_match($collision_2, $username) || preg_match($collision_3, $username))
+    {
+        // redirect user to the RestrictRedirect page
+        $wgOut->redirect($CASAAuth["RestrictRedirect"]);
+        return true;
+    }

+    // Get MediaWiki user
+    $u = User::newFromName($username);
+
+    }
```

While I have only tested the CASAAuth extension, I fear that this security bug may be more widespread, so I've only reported this issue to other members of the technical team I am a part of at work, and to you.

Does your team generally assist in coordination with multiple extension maintainers, or should I try to contact every maintainer of MW SSO login extensions? I figure that if that there isn't any coordination, that some maintainers may release before others, leading to insight by malicious users.

Thanks for your help. :) Andrew

Related Objects

Mentions

Mentioned In

🔗 #63040: Write and send supplementary release announcement for extensions and skins with security patches (1.31.11/1.35.1)

🔗 #56341: Obtain CVEs for 1.31.9/1.34.3/1.35.0 security releases

🔗 #56342: Write and send supplementary release announcement for extensions and skins with security patches (1.31.9/1.34.3/1.35.0)

Mentioned Here

🔗 #63040: Write and send supplementary release announcement for extensions and skins with security patches (1.31.11/1.35.1)

🔗 #63002: Release MediaWiki 1.31.11/1.35.1

🔗 #56342: Write and send supplementary release announcement for extensions and skins with security patches (1.31.9/1.34.3/1.35.0)

🔧 Sudozero created this task. 2020-09-21 20:57:56 (UTC+0)

👤 Restricted Application added a subscriber: **Aklapper**. · View Herald Transcript 2020-09-21 20:57:58 (UTC+0)

👤 Reedy added a project: **MediaWiki-Authentication-and-authorization**. 2020-09-22 01:42:20 (UTC+0)

👤 Reedy added a subscriber: **Reedy**. 2020-09-22 01:45:26 (UTC+0)

FWIW, any amounts of underscores/spaces get replaced with one. So your mitigation isn't sufficient :)

> echo ( User::getCanonicalName( "Foo\_\_\_\_bar" ) );  
Foo bar

```
> echo ( User::getCanonicalName( "Foo    bar" ) );
Foo bar
```

**Reedy** updated the task description. ([Show Details](#)) 2020-09-22 02:25:31 (UTC+0)

**Sudozero** added a comment. 2020-09-22 17:46:22 (UTC+0)

Thanks for the tip. :) I think that the regex for `/___/` covers it, since three or more underscores in a row should still be matched by two of them. The same goes for the spaces. I don't know if this is the best way to handle this, but it seems like an okay stop-gap measure.

FWIW, we seem to have far more users with one or more spaces in their names (~1K) than we do people who have underscores (~100).

Of course if people are blocked from logging in with their account, they will need to contact me or another admin at my org so we can change their user name, which would allow them to log into our wiki again. Most of those people are unlikely to be actually using their account or visiting our wiki, so there won't be all that many blocks in effect. The main concern is to stop someone from maliciously logging in as admin, or other wiki users.

Thanks! :)

**sbassett** added a subscriber: **sbassett**. 2020-09-22 20:14:41 (UTC+0)

@Sudozero -

*Thanks for the tip. :) I think that the regex for `/___/` covers it, since three or more underscores in a row should still be matched by two of them. The same goes for the spaces. I don't know if this is the best way to handle this, but it seems like an okay stop-gap measure.*

As [@Reedy](#) implied above, I'd be cautious when using regular expressions to sanitize any data as they can become complicated and unwieldy very quickly. I'd also recommend writing some unit tests for CASAuth ([mediawiki primarily uses phpunit for php code](#)) to ensure you have all of your expected edge cases covered.

*Does your team generally assist in coordination with multiple extension maintainers, or should I try to contact every maintainer of MW SSO login extensions? I figure that if that there isn't any coordination, that some maintainers may release before others, leading to insight by malicious users.*

We can help out a bit, but we mainly focus on [Wikimedia-deployed](#) and [bundled](#) extensions and skins. For any announcement to potential users of CASAuth, I would probably start with [the wikiapiary list of wikis using the extension](#) and try to find contacts for those projects to supplement any personal knowledge of users you may have. Once a patch is written and backported to any relevant branches, we can include this task within our tracking task for quarterly security announcements ([T256342](#)) which are sent out to various mediawiki-related mailing lists ([here's a recent example](#)). For code with a canonical repo at github, we can reference any relevant pull requests.

- sbassett** mentioned this in ~~T256342- Write and send supplementary release announcement for extensions and skins with security patches (1.31.9/1.34.3/1.35.0)~~. 2020-09-22 21:44:26 (UTC+0)
- sbassett** mentioned this in ~~T256341- Obtain CVEs for 1.31.9/1.34.3/1.35.0 security releases~~. 2020-09-23 21:19:30 (UTC+0)
- sbassett** moved this task from **Incoming** to **Watching** on the **Security-Team** board. 2020-09-28 15:21:53 (UTC+0)
- sbassett** mentioned this in ~~T263040- Write and send supplementary release announcement for extensions and skins with security patches (1.31.11/1.35.4)~~. 2020-11-17 22:35:13 (UTC+0)

**sbassett** added a comment. 2020-12-01 21:49:21 (UTC+0)

@Sudozero -

just wanted to check in and see if there was any additional progress made on this effort. Thanks.

**Sudozero** added a comment. 2020-12-02 00:51:04 (UTC+0)

Hi [@sbassett](#)

I emailed site owners about the issue on 2020-11-23, by using mostly technical support contact addresses on their parent sites, but haven't heard back from any of them so far.

I have the patch I quoted earlier working on the wikis that I manage, aside from additional symbols that I need to investigate. However I suspect that not all CAS servers will allow the same set of characters in their user names. I haven't written any tests for the changes. Because I haven't heard back from any admins, I haven't gotten any help for coding a patch.

I am up for at least making an issue with a sample patch on the GitHub project page, but I'm not sure about an official pull request, since that would likely need more work. In my email to site admins, I mentioned that I wanted to give them a head start on patching their systems before I create a GitHub issue upstream, so depending on the timing of your security newsletter, I could create an issue before then, if that sufficient for your announcement.

I know that I have not gone above and beyond regarding this report and patch, but if you have any suggestions, I'm open to hear them.

I know that I mentioned this to an extent earlier, but I'm concerned that others who don't use CASAuth may also be impacted, because this type of vulnerability seems to be based on a trivial assumption about how user names are processed by MW. I don't want malicious actors to attack sites using popular SSO extensions too.

Thanks for checking in. :)

Andrew

**sbassett** added a comment. 2020-12-02 22:27:04 (UTC+0)

In ~~T263498#6661625~~, [@Sudozero](#) wrote:

*I emailed site owners about the issue on 2020-11-23, by using mostly technical support contact addresses on their parent sites, but haven't heard back from any of them so far.*

Great, that's really all that can be done in most cases. Extreme hand-holding only leads to frustration :)

*I am up for at least making an issue with a sample patch on the GitHub project page, but I'm not sure about an official pull request, since that would likely need more work. In my email to site admins, I mentioned that I wanted to give them a head start on patching their systems before I create a GitHub issue upstream, so depending on the timing of your security newsletter, I could create an issue before then, if that sufficient for your announcement.*

Ok, you could likely guard this new functionality with a config variable. Then other users could enable/disable the functionality based upon their needs.

*I know that I have not gone above and beyond regarding this report and patch, but if you have any suggestions, I'm open to hear them.*

Oh, I think you've done fine. For any security issue like this, we typically want to make best efforts around addressing the issue, writing a correct patch, informing relevant users/operators and then disclosing the issue and backporting as necessary.

*I know that I mentioned this to an extent earlier, but I'm concerned that others who don't use CASAuth may also be impacted, because this type of vulnerability seems to be based on a trivial assumption about how user names are processed by MW. I don't want malicious actors to attack sites using popular SSO extensions too.*

Yes, that's likely something for the maintainers of other SSO extensions to test and confirm. Hopefully addressing and eventually disclosing an issue like this will result in further action on that front.

**Sudozero** added a comment. 2020-12-06 21:10:39 (UTC+0)

Is there a rough timeline of when the security announcement will go out? I'd like to get the patch ready before then. Thanks : )

sbassett added a comment. Edited · 2020-12-08 21:50:26 (UTC+0)

@Sudozero - I would guess the current security release ( **T263802** ) will happen sometime over the next two or so weeks? @Reedy might have a more specific date in mind. Though if you need more time with any of this, we can always include this within the next supplemental announcement.

Reedy added a comment. 2020-12-08 22:47:30 (UTC+0)

In **T263498#6677701**, @sbassett wrote:

@Sudozero - I would guess the current security release ( **T263802** ) will happen sometime over the next two or so weeks? @Reedy might have a more specific date in mind. Though if you need more time with any of this, we can always include this within the next supplemental announcement.

Planning for 2020-12-16 or 2020-12-17 so they're out comfortably before Christmas

Sudozero added a comment. 2020-12-13 20:45:28 (UTC+0)

Okay, I worked on improving my patch, and so I'll put it here for now.

```
--- CASAuth.php.orig 2019-10-10 16:56:19.679560769 -0400
+++ CASAuth.php 2020-12-13 15:35:33.398765653 -0500
@@ -113,6 +113,13 @@
         // Get username
         $username = casNameLookup($phpCAS::getUser());

+        // casNameLookup() says name is invalid
+        if (is_null($username)) {
+            // redirect user to the RestrictRedirect page
+            $wgOut->redirect($CASAuth["RestrictRedirect"]);
+            return true;
+        }

+        // If we are restricting users AND the user is not in
+        // the allowed users list, lets block the login
+        if($CASAuth["RestrictUsers"]==true
```

And in CASAuthSettings.php, replace the default hook with:

```
function casNameLookup($username) {

    // Some special characters are automatically trimmed from user names when
    // logging in. For instance if a user logs in with the CAS account name
    // "_admin_user", they would normally be logged into MediaWiki as
    // "Admin user". This code blocks certain combinations of underscores and
    // spaces in user names. A patch to CASAuth.php is also required.

    // Normally, both "admin_user" and "admin user" both log in as "Admin user".
    // Allow isolated underscores rather than spaces (true or false):
    $preferUnderscore = true;

    $collisions = [ "/^_/", "/_$/", "/^ /", "/ $/", "/ /", "/_ /", "/_ _/" ];
    $collisions[] = $preferUnderscore ? " / " : " /_ /";

    foreach ($collisions as $collision) {
        if(preg_match($collision, $username)) {
            unset($collision);
            // reject user name
            return null;
        }
    }
    unset($collision);

    // user name checks out
    return $username;
}
```

Sudozero added a comment. 2020-12-14 19:42:54 (UTC+0)

@sbassett Is it okay for me to post the issue or merge request on GitHub a few days before the planned security announcement email goes out? I can shrink that gap if needed. Thanks.

sbassett added a comment. 2020-12-14 19:48:32 (UTC+0)

In **T263498#6689905**, @Sudozero wrote:

@sbassett Is it okay for me to post the issue or merge request on GitHub a few days before the planned security announcement email goes out? I can shrink that gap if needed. Thanks.

If you're happy with the patch and any recent disclosure/communication efforts to relevant users, I'd recommend merging the patch as soon as possible. For extensions like this, which aren't bundled for MediaWiki security releases, the sooner a fix and disclosure can occur, the better. The supplemental announcement can be viewed as something to help increase visibility even further, but does not need to serve as the primary means of disclosure/communication for any security issue.

Sudozero added a comment. 2020-12-15 01:07:33 (UTC+0)

I created a pull request here: <https://github.com/CWRUChielLab/CASAuth/pull/10>

I also emailed others who have made commits to forks of that repo according to GitHub's network feature.

sbassett added a comment. 2020-12-15 16:53:16 (UTC+0)

In **T263498#6690696**, @Sudozero wrote:

I created a pull request here: <https://github.com/CWRUChielLab/CASAuth/pull/10>

I also emailed others who have made commits to forks of that repo according to GitHub's network feature.

Sounds good. Once you merge that request, I'll include this issue within the next supplemental announcement ( **T263810** ), likely due out next week.


Sudozero added a comment. 2020-12-17 18:12:42 (UTC+0)

@sbassett The patch was merged!


<https://github.com/CWRUChielLab/CASAuth/pull/10>

<https://github.com/CWRUChielLab/CASAuth>

 **sbassett** awarded a token. 2020-12-17 21:16:36 (UTC+0)

 **Sudozero** added a comment. 2020-12-19 15:29:36 (UTC+0)


Should I make a CVE for this security issue?

 **sbassett** added a comment. 2020-12-20 19:28:50 (UTC+0)

In [T263498#6703973](#), @Sudozero wrote:  
*Should I make a CVE for this security issue?*


Feel free, otherwise I can tomorrow. [This is the form](#) I typically use.

→ **sbassett** triaged this task as *Low* priority. 2020-12-20 19:29:05 (UTC+0)

 **Sudozero** added a comment. 2020-12-20 19:29:14 (UTC+0)

For the sake of reference, the user name normalization appears to occur in the `splitTitleString` method, in `includes/title/MediaWikiTitleCodec.php`. There are additional special Unicode characters that are stripped out there.

The characters that get stripped there include bidirectional override characters: `/\xE2\x80[\xE8\x8F\xAA-\xAE]/`. The following pattern is replaced with `'_':/[_xA0{x1680}{x180E}{x2000}-\x{200A}{x{2028}}{x{2029}}{x{202F}}{x{205F}}{x{3000}}]+/u`.


 **Sudozero** added a comment. 2020-12-20 22:10:23 (UTC+0)

[@sbassett](#)

I added another pull request to cover more edge cases: <https://github.com/CWRUChielLab/CASAuth/pull/11>

As for creating the CVE, it might best if you create it.

Thanks :)


 **Sudozero** added a comment. 2020-12-20 22:18:04 (UTC+0)

That commit was merged upstream.


 **sbassett** added a comment. 2020-12-21 20:55:16 (UTC+0)

In [T263498#6705146](#), @Sudozero wrote:  
*As for creating the CVE, it might best if you create it.*

I've now requested a CVE for this issue. The supplemental release should go out tomorrow or Wednesday. Thanks for working on this!


 **Sudozero** added a comment. 2020-12-22 02:10:11 (UTC+0)


[@sbasset](#) Thank you for your guidance and for your help! :)

 **sbassett** renamed this task from *Logins to MW with at least one SSO client extension allows masquerading as another user* to *Logins to MW with at least one SSO client extension allows masquerading as another user (CVE-2020-35623)*. 2020-12-22 20:51:51 (UTC+0)

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

 **sbassett** closed this task as *Resolved*. 2020-12-22 20:53:55 (UTC+0)

 **sbassett** assigned this task to **Sudozero**.

 **sbassett** moved this task from **Watching** to **Our Part Is Done** on the **Security-Team** board.

 **sbassett** added a project: **Vuln-Authn/Session**. 2021-03-16 21:39:27 (UTC+0)