Use of Wrong Operator in String Comparison in hestiacp/hestiacp

0

✓ Valid Reported on Sep 10th 2021

✍️ Description

`$_SESSION["token"]` is a csrf token which is a md5 hash generated based on system time.
It has been discovered that `$_SESSION["token"]` compares with `$_GET["token"]` using comparison operator `!=` in file `index.php` . This might cause unexpected behavior due to type juggling.
It is possible to bypass the CSRF token by using magic hash attack, and leveraged to perform CSRF attack.

Remediation

Use `!==` instead.

Occurrences

index.php L9 index.php L305

References

- Comparison Operators

CVE
CVE-2021-3797
(Published)

Vulnerability Type
CWE-597: Use of Wrong Operator in String Comparison

Severity
Medium (4.8)

Affected Version
*

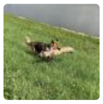
Visibility
Public

Status
Fixed

Found by

Viky
@vikychoi
unranked

Fixed by



Jaap Marcus
@jaapmarcus
maintainer

This report was seen 478 times.

We have contacted a member of the **hestiacp** team and are waiting to hear back a year ago

Viky submitted a patch a year ago

Viky submitted a patch a year ago

Jaap Marcus validated this vulnerability a year ago

Viky has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

Jaap Marcus a year ago Maintainer

Patch is not complete there are more files affected. I will go over all files and fix the issues if you don't mind.

Chat with us

Jaap Marcus marked this as fixed with commit fc68ba a year ago

Jaap Marcus has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

Viky a year ago Researcher

@admin can I have a cve?

Jamie Slome a year ago Admin

CVE published! 🎊

CVE-2021-3797

Jaap Marcus a year ago Maintainer

@admins

I think a CVE is a bit over done:

Token is not generated by MD5 but

https://github.com/hestiacp/hestiacp/blob/ba84b5ad93dc5f33894931d7a7350684a85e7acf/web/inc/main.php#L87-L93

So in the rare cases where $_SESSION['token'] is empty what should never happen or in the rare cases where the generated session token would generate a valid number.

With a 16 char length random string the chances would be very small.

I do agree the suggested improvements are correct and valid but practical use it would be almost impossible to abuse it.

Jamie Slome a year ago Admin

Hello Jaap, 👋

We published the CVE as you indicated that this was a valid security issue and agreed with the contents of the report (CVSS etc.).

In the future, if this is not the case, please let us or the researcher aware by invalidating the report.

-- Jamie

myvesta a year ago

I must correct @Jaap.
He said: "should never happen or in the rare cases"
That's not correct.
Correct is that this issue is IMPOSSIBLE to happen.

Even session_id is number '==' will correctly do it's job.

'==' will fail only if session_id is '0' and token is empty.
Since session_id will never be '0', this so-called "issue" is impossible to happen.

So, to be precisely, you have CVE that is impossible to produce.
I will not confirm this issue in VestaCP and myVesta.

myvesta a year ago

Looks like I was wrong,
"0x" will match empty value.
I will do additional check.
I appologize for my previous comment.

Jaap Marcus a year ago Maintainer

See POC below...

```
<?php
$i = 0;
while(1 == 1){
$i++;
$token = bin2hex(file_get_contents('/dev/urandom', false, null, 0, 16));
if(substr($token,0,2) == '0e'){
    echo $token."\r\n";
        if($token == 0){
            echo "Match found";
            echo $token ."\r\n";
            echo $i."\r\n";
            die();
        }
}
}
?>
```

```
root@dev:~# php poc.php
0eb90ab7a008d65842b9996f9d32043e
Match found after: xx attempts...
317
634
175
27
33
```

I also noticed the issue was against an old branch master. We current use main...

myvesta a year ago

if ($token == '')
is not the same as
if ($token == 0)
but anyway, they have a point, someone can use 0 as token

myvesta a year ago

After additional check, looks like I was even correct.
we can not use example of
if ($token == 0)
because this is integer on right side, and $_GET['token'] will never be threated as integer.

So, at the end, this issue can not be exploited.

But however, I will change == to === in my fork, just to satisfy good code practice.

Sign in to join this conversation