ATTACK DEFENSE

by PentesterAcademy

Name	Bundler Audit: Scanning vulnerabilities in Ruby Gems
URL	https://www.attackdefense.com/challengedetails?cid=2154
Type	DevSecOps Basics: Software Component Analysis

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Challenge Description

The Bundler Audit will audit the Gemfile.lock file to check for any vulnerable gems installed.

A Kali CLI machine (kali-cli) is provided to the user with bundler-audit installed on it. The source code for two sample applications is provided in the home directory of the root user.

Objective: Use the bundler-audit utility to find vulnerable gems!

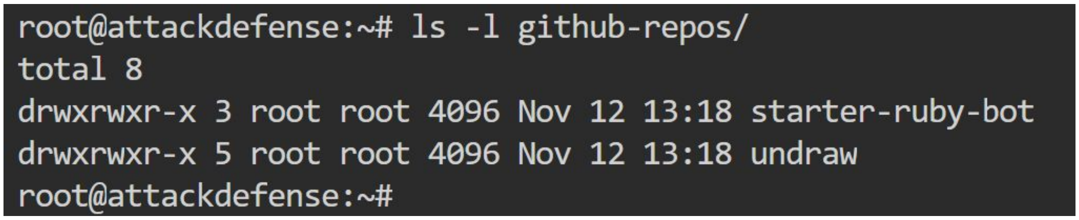
Instructions:

- The source code of applications is provided at /root/github-repos

Solution

Step 1: Check the provided applications.

Command: ls -l github-repos/



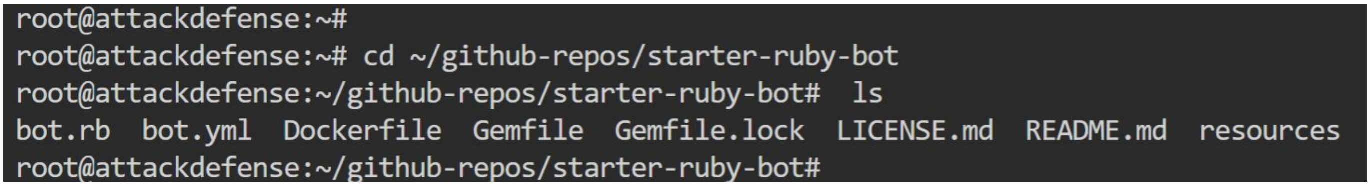
```
root@attackdefense:~# ls -l github-repos/
total 8
drwxrwxr-x 3 root root 4096 Nov 12 13:18 starter-ruby-bot
drwxrwxr-x 5 root root 4096 Nov 12 13:18 undraw
root@attackdefense:~#
```

We will take one example at a time and run the tool on that.

Example 1: starter-ruby-bot

Step 1: Navigate to the starter-ruby-bot directory and check it’s content.

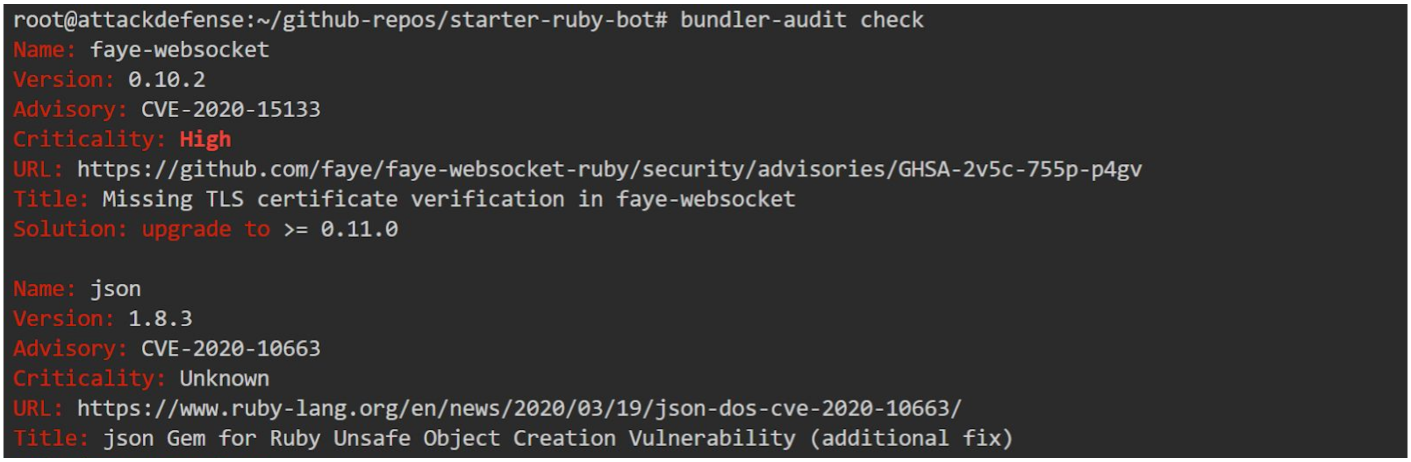
Commands:
cd ~/github-repos/starter-ruby-bot
ls



```
root@attackdefense:~#
root@attackdefense:~# cd ~/github-repos/starter-ruby-bot
root@attackdefense:~/github-repos/starter-ruby-bot#  ls
bot.rb  bot.yml  Dockerfile  Gemfile  Gemfile.lock  LICENSE.md  README.md  resources
root@attackdefense:~/github-repos/starter-ruby-bot#
```

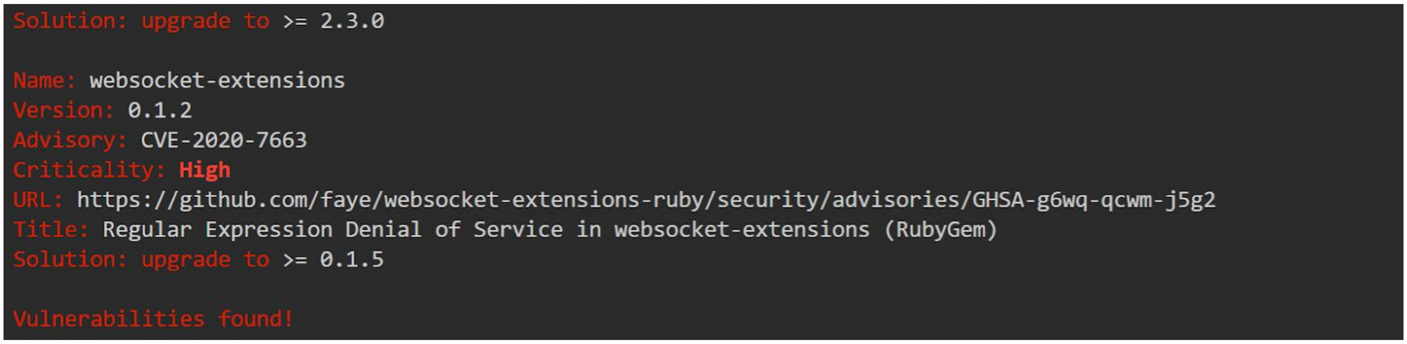
Step 2: Run the bundler-audit on the project.

Command: bundler-audit check



```
root@attackdefense:~/github-repos/starter-ruby-bot# bundler-audit check
Name: faye-websocket
Version: 0.10.2
Advisory: CVE-2020-15133
Criticality: High
URL: https://github.com/faye/faye-websocket-ruby/security/advisories/GHSA-2v5c-755p-p4gv
Title: Missing TLS certificate verification in faye-websocket
Solution: upgrade to >= 0.11.0

Name: json
Version: 1.8.3
Advisory: CVE-2020-10663
Criticality: Unknown
URL: https://www.ruby-lang.org/en/news/2020/03/19/json-dos-cve-2020-10663/
Title: json Gem for Ruby Unsafe Object Creation Vulnerability (additional fix)
```



```
Solution: upgrade to >= 2.3.0

Name: websocket-extensions
Version: 0.1.2
Advisory: CVE-2020-7663
Criticality: High
URL: https://github.com/faye/websocket-extensions-ruby/security/advisories/GHSA-g6wq-qcwm-j5g2
Title: Regular Expression Denial of Service in websocket-extensions (RubyGem)
Solution: upgrade to >= 0.1.5

Vulnerabilities found!
```

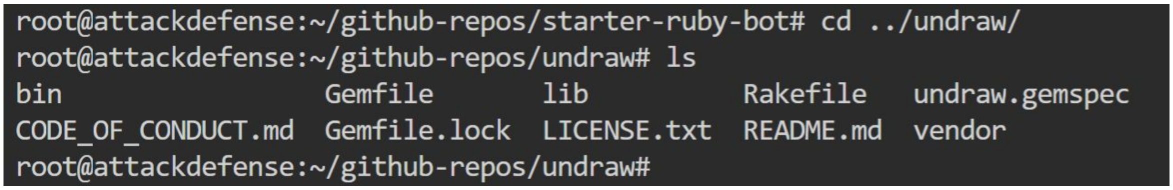
Issues Detected

- Faye-websocket version 0.10.2 is vulnerable to CVE-2020-1533
- Json gem version 1.8.3 is vulnerable to CVE-2020-10663
- Websocket-extensions version 2.3.0 or below is vulnerable to CVE-2020-7663

Example 2: Undraw

Step 1: Change to the undraw directory and check it's content.

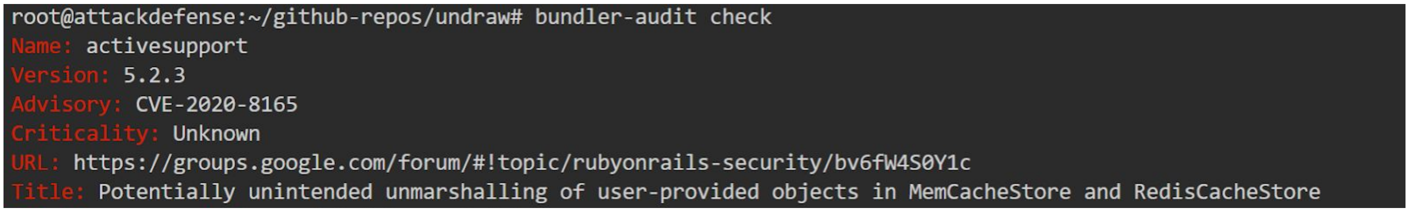
Commands:
cd ../undraw/
ls



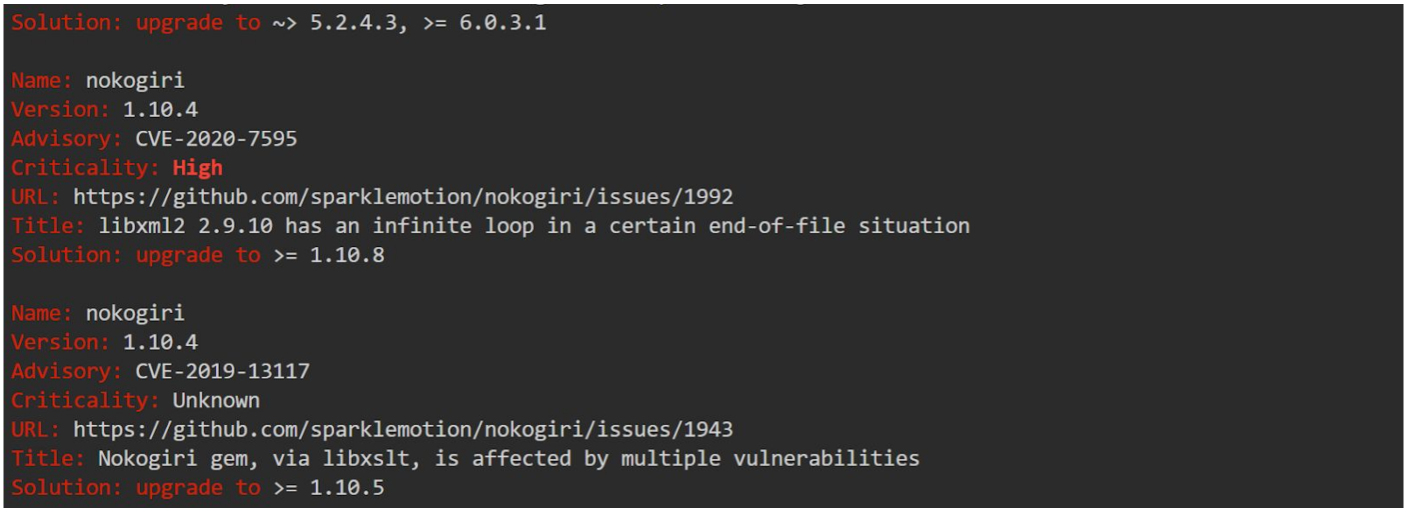
```
root@attackdefense:~/github-repos/starter-ruby-bot# cd ../undraw/
root@attackdefense:~/github-repos/undraw# ls
bin                 Gemfile       lib          Rakefile   undraw.gemspec
CODE_OF_CONDUCT.md  Gemfile.lock  LICENSE.txt  README.md  vendor
root@attackdefense:~/github-repos/undraw#
```

Step 2: Run the bundler-audit on the project.

Command: bundler-audit check



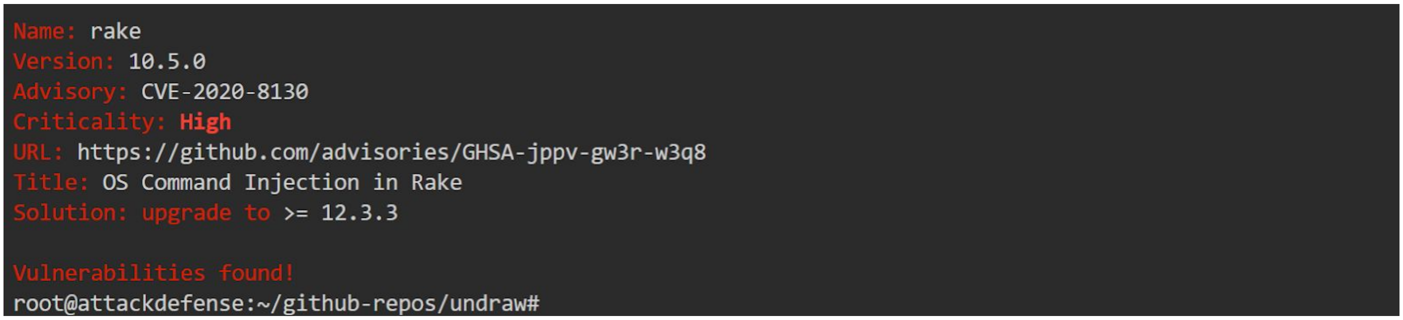
```
root@attackdefense:~/github-repos/undraw# bundler-audit check
Name: activesupport
Version: 5.2.3
Advisory: CVE-2020-8165
Criticality: Unknown
URL: https://groups.google.com/forum/#!topic/rubyonrails-security/bv6fW4S0Y1c
Title: Potentially unintended unmarshalling of user-provided objects in MemCacheStore and RedisCacheStore
```



```
Solution: upgrade to ~> 5.2.4.3, >= 6.0.3.1

Name: nokogiri
Version: 1.10.4
Advisory: CVE-2020-7595
Criticality: High
URL: https://github.com/sparklemotion/nokogiri/issues/1992
Title: libxml2 2.9.10 has an infinite loop in a certain end-of-file situation
Solution: upgrade to >= 1.10.8

Name: nokogiri
Version: 1.10.4
Advisory: CVE-2019-13117
Criticality: Unknown
URL: https://github.com/sparklemotion/nokogiri/issues/1943
Title: Nokogiri gem, via libxslt, is affected by multiple vulnerabilities
Solution: upgrade to >= 1.10.5
```



```
Name: rake
Version: 10.5.0
Advisory: CVE-2020-8130
Criticality: High
URL: https://github.com/advisories/GHSA-jppv-gw3r-w3q8
Title: OS Command Injection in Rake
Solution: upgrade to >= 12.3.3

Vulnerabilities found!
root@attackdefense:~/github-repos/undraw#
```

Issues Detected:

- Activesupport version 5.2.3 is vulnerable to CVE-2020-8165
- Nokogiri version 1.10.4 is vulnerable to CVE-2020-7595, CVE-2019-13117
- Rake version 10.5.0 is vulnerable to CVE-2020-8130

Learnings

Perform Software Component Analysis using the bundler-audit tool.

References:

- Starter Ruby Bot (https://github.com/salted/starter-ruby-bot)
- Undraw (https://github.com/mkhairi/undraw.git)