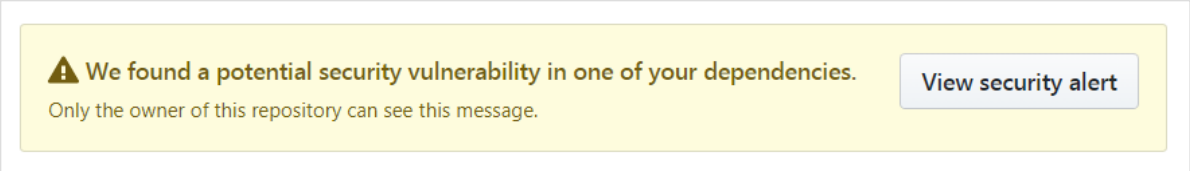
Updating Dependencies

Posted on **March 18, 2020**

There are things you don't want to see when you review your code. Here's one:



— Potential Security Vulnerability Warning Alert

What's worse, this message was hard to track down. There was no way (that I found) to have GitHub tell me about all the code that was affected. I'd have to check each one manually until I had found and fixed them all.

I had several things going for me. The alerts were all caused by one thing; the discovery of a bug in the "rake" gem. Here's one version of the bug report I found:

> *It was discovered that Rake incorrectly handled certain files. An attacker could use this issue to possibly execute arbitrary commands.*
>
> USN-4295-1: RAKE VULNERABILITY

Now, in theory, one could do an in depth analysis of this bug and conclude that it has no impact on your code. I do not recommend this approach. I would not have confidence that I could see all the devious ways this fault could be exploited. Further, I know I cannot possibly predict every move by the criminals. That is why I will now focus on ensuring that my code specifies a version of rake that does not have this defect. That is we need to specify versions of rake with version greater than or equal to 12.3.3.

I write Ruby Gems and for those gems, dependencies are specified in the *<gem_name>.gemspec* file. Let's see some rake dependency entries and see how they stack up to our security

requirement:

spec.add_development_dependency “rake”, “~> 10.0”

This was the dependency entry that generated my alert messages. To translate into plain English, it says that any version of rake is OK, so long as it was version 10.something. The last such release was 10.5.0 which clearly has the bug and thus generates a warning. This entry clearly must be replaced.

Now many of my Ruby Gems had the following entry that did not result in a warning, but still leaves something to be desired:

spec.add_development_dependency “rake”, “~> 12.0”

This entry says to accept any version of rake so long as it was version 12.something. The last such release was 12.3.3 which is clear and free of the bug! So, yes, this is much better, but it suffers from the fact that it limits rake to version 12. Version 13 is already out and forcing the use of an older version may not be a good idea. I chose to replace these entries as well.

So, we now examine the simplest and most permissive specification:

spec.add_development_dependency ‘rake’

This translates to anythings goes, but use the very latest if it’s not already installed. This is very simple, it generated no warnings, and it can get the latest. It will also settle for much older versions, including those with the bug. This too must be updated.

This is the entry recommended for use by the alert:

spec.add_development_dependency “rake”, “>= 12.3.3”

This translates to “Anything from 12.3.3 or later”. This ensures that we will not use older versions of (buggy) code, and that we won’t exclude newer versions of code either.

Now there may be legitimate reasons to stick to older code. I know there are working web sites using ancient versions of rails running on equally old versions of ruby. Even so, there are (too many to list here) real risks to running obsolete code. If at all possible, keep up-to-date. It may require some effort, but it is worth it. Test your code with newer versions *before* pushing them out into the wild. If problems/issues arise, fix them and update your dependencies. This usually can be accomplished in a modest amount of time. If it can’t then you need to lock in the obsolete code dependencies and take a serious look at re-engineering your application.

In the end, everything needs maintenance. The reason we all avoid gems that have gone many years with no updates is that we know that nobody is tending house. I was caught by surprise and it is hard to anticipate what the next disruption will be, but by defensively coding, the risk can be reduced.

Finally let me take a moment to list which of my gems needed to be updated due to this issue. If you have a current, safe version of rake installed, you need do nothing even if you are using these gems: *composite_rng, connect_n_game, counted_cache, fibonacci_rng, format_engine, format_output, full_clone, full_dup, fully_freeze, in_array. insouciant, lexical_analyzer, make_gem, mini_erb, mini_readline, mini_term, mysh, parse_queue, pause_output, ruby_sscanf, safe_clone, safe_dup, test65,* and *vls*. In addition, *fOOrth, games_lessons*, and *rctp* were updated but the changes are currently part of unreleased code.

Yours Truly

Peter Camilleri (aka Squidly Jones)

This entry was posted in **Hidden Traps**, **Ruby** by **Peter Camilleri**. Bookmark the **permalink [http://teuthida-technologies.com/?p=1852]** .