Pulp - Story #8236

Enable users to protect against mixed internal / external repository package name collision attacks

02/10/2021 06:24 AM - dalley

Status:	NEW	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:			
Platform Release:		**Tags:**	Wishlist
Groomed:	No	**Sprint:**	
Sprint Candidate:	No	**Quarter:**	

Description

https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610

TL;DR the names of internal-registry-only packages can frequently leak outside of a company. Attackers can add them to a public registry and subsequent syncs of that public registry by repo management tools like Pulp and Artifactory will mix the good-internal and malicious-external packages together. This can be used to install malicious code within corporate networks when developers or CI services install these dependencies.

A concrete example:

> For instance, the main culprit of Python dependency confusion appears to be the incorrect usage of an “insecure by design” command line argument called --extra-index-url. When using this argument with pip install library to specify your own package index, you may find that it works as expected, but what pip is actually doing behind the scenes goes something like this:
>
> - Checks whether library exists on the specified (internal) package index
> - Checks whether library exists on the public package index (PyPI)
> - Installs whichever version is found. If the package exists on both, it defaults to installing from the source with the higher version number.
>
> Therefore, uploading a package named library 9000.0.0 to PyPI would result in the dependency being hijacked in the example above.
>
> Although this behavior was already commonly known, simply searching GitHub for --extra-index-url was enough to find a few vulnerable scripts belonging to large organizations — including a bug affecting a component of Microsoft’s .NET Core. The vulnerability, which may have allowed adding backdoors to .NET Core, was unfortunately found to be out of scope in the .NET bug bounty program.
>
> ...
>
> JFrog Artifactory, a piece of software widely used for hosting internal packages of all types, offers the possibility to mix internal and public libraries into the same “virtual” repository, greatly simplifying dependency management. However, multiple customers have stated that Artifactory uses the exact same vulnerable algorithm described above to decide between serving an internal and an external package with the same name. At the time of writing, there is no way to change this default behavior.

We've got the same problem as Artifactory, there's no way to distinguish between internal package namespaces and external ones. If you just copy them into the same repository, you're left vulnerable to this attack.

I'm not completely sure how we would resolve this. Maybe take the Foreman concept of "content views" and formalize it so that various repositories are layered on top of each other, but with the ability to detect package name collisions between repos?

History

#1 - 02/10/2021 06:34 AM - dalley

- Description updated

#2 - 02/22/2021 03:46 PM - dalley

Here's an architectural change that I think would make this possible.

Instead of syncing arbitrary remotes into arbitrary repositories, every remote would be paired with exactly one repository and would remain an exact clone of the remote repository at all times.

To do content management, we would have meta-repositories, where many repositories could be combined together, and one repository would be automatically created for "user changes", which would **always** take priority over the others. So if a user adds an package to the internal registry it will never, ever be shadowed by an external package with the same name. We could also detect that situation and provide advance warning that it occurred.

I believe would also completely eliminate the need for Katello to handle "content views", because Pulp would be fully capable of handling those scenarios.