The Python ecosystem is messed up and here’s why...

Aadesh M. Bagmar∗and Katherine Sullivan
University of Maryland
College Park, MD, 20740

The popularity of Python has risen rapidly over the past 15 years. It is a major language in some of the most exciting technologies today. This popularity has led to a large ecosystem of third party packages available via the pip package registry which hosts more than 200,000 packages. These third party packages can be reused by simply importing the package after installing using package managers like pip. The ease of reuse of third party software comes with security risks putting millions of users in danger. In this project, we study the ecosystem to analyze this threat. We scraped entire pypi.org for this study.

The tightly knit ecosystem of Python has a lot of weak spots that we highlight in our project. First, we demonstrate a simple way to exploit the core pip architecture which allows a user to get a reverse shell to any system installing the package. Then, we analyze the dependency graph and evaluate the average number of third party packages a user trusts while installing a package. We then evaluate the impact of compromising a package and assign a blast radius to popular python packages.

Most attacks are possible only if users install malicious packages. We thus try to analyze and evaluate different methods used by attackers to force incorrect downloads like typo-squatting. We then discuss an attack that happened in the real-world with the jellyfish python package in December 2019 which stole users SSH keys. Our experiments also led us to The Guardian Project which uses methods used by attackers as a defense to protect the popular packages. Analysis of the Guardian Project helps us compute a lower bound on number of times users download incorrect packages.

Lastly, our data enables us to identify potential software license violations due to code reuse by importing packages. We aim to alert maintainers of the packages of these violations and analyze their response in the future.

In our extensive report (not covered in the poster), we suggest ways to secure pip, analyze the minimum number of packages/maintainer accounts an attacker needs to compromise to get hold of the entire ecosystem and analyze the cycle of vulnerability detection and fixing in popular packages across releases.

∗Aadesh Bagmar is a member of the SEAM lab (Software Engineering Lab at Maryland) and is supported by Office of Naval Research under contract N000141812767