pip install exfil

DATA EXFILTRATION AND C2 USING PUBLIC PACKAGE REPOSITORIES

Pat H
/path/to/file

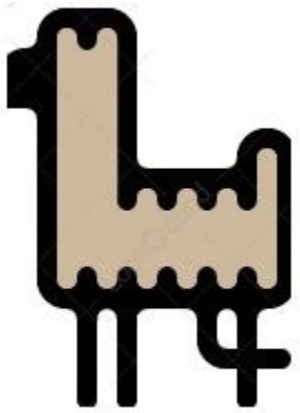
Intro

- Trader of Python code for money
- Three things kicked off this research:

1. Home server runs Continuous Integration (CI) for my silly software projects
   - This automatically downloads and installs packages from Pypi.Python.org
   - Website reputation: I trust it enough!
   - Led to “whitelisting” (read: ignoring) SSL traffic as benign, if during expected hours

2. Vincenzo Iozzo – Bsides Zurich 2018 – “Offense: R.I.P. good times”
   - “In a modern network, defense will have the upper hand”
   - “One new possible vector (of many) – Repos/CDN”

3. King Kuzco



Software repositories

- Most programming languages and OSs have a central server people can upload libraries to
  - PyPi, NuGet, Rubygems, NPM, etc.
  - Apt, Yum, AUR
  - Github for code and “raw” files
  - Typically 1000s of packages, millions of downloads
  - Updates typically involve downloading and installing any dependencies automatically
- While packages not necessarily trusted, repositories generally are:
  - Well known, good reputation websites with trusted SSL certificates, etc.
  - Downloaded (and uploaded) using HTTPS
  - Normal for a company build server to be communicating to these systems

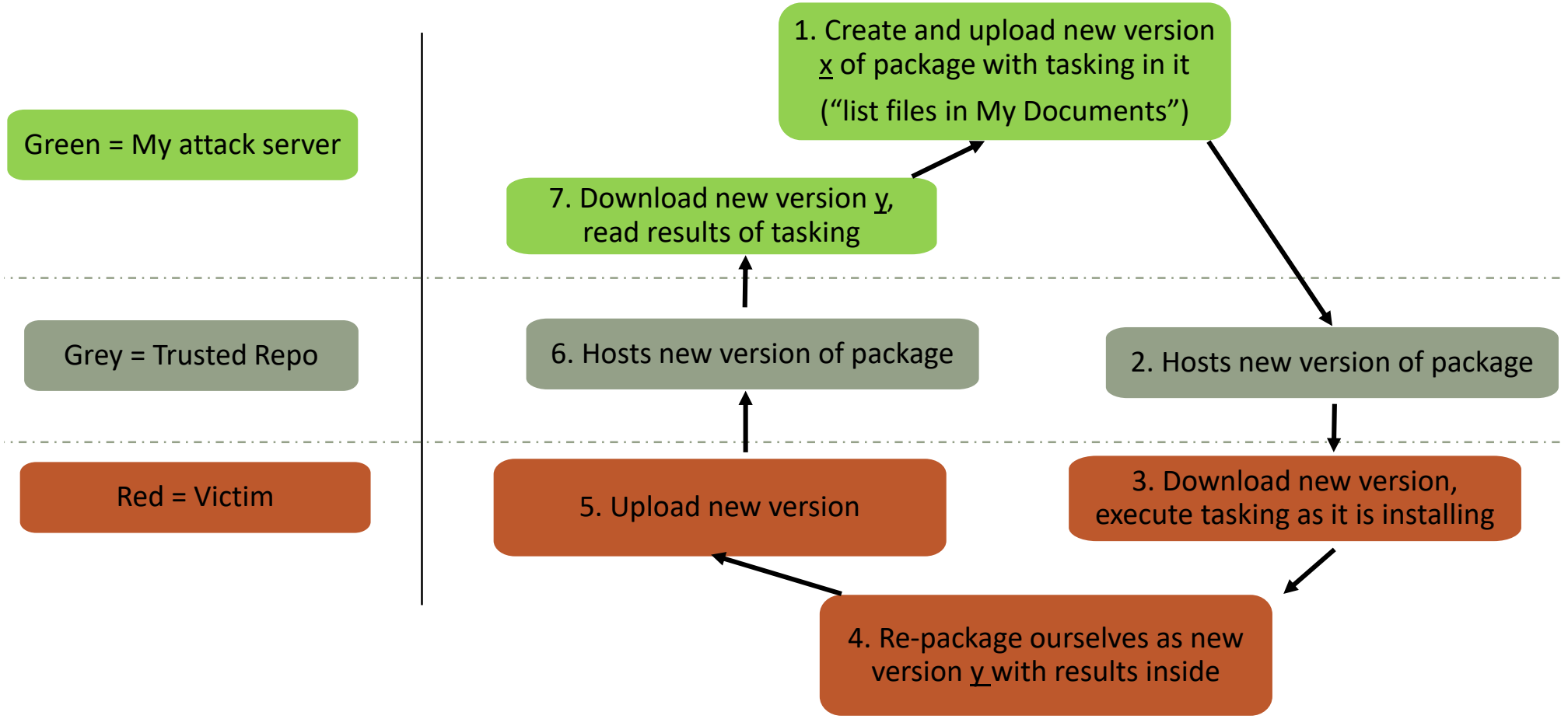
Research Question:

- So, say you are a company with a server that downloads and builds software:
- And I have either:
  1. Gotten access to the server and can run code on it; OR
  2. I've compromised a package you depend on (or a dependency of a dependency of a depe…)
     - Aka dependency supply-chain attack

A. Can I secretly exfiltrate data from your network, hiding in the noise of all the other traffic going to and from these 'trusted' servers?

B. What protections do the repositories have to prevent this abuse?

Plan of attack – Self publishing package



Criteria of success

For each repository, can we:

1. Create and upload packages anonymously
   - Used 10Minutemail for email validation
2. Rapidly and programmatically build and re-publish new versions
   - Fast turnaround is key
   - Also need to be able to re-package and re-publish ourselves
3. Enable code-execution at install time
   - For scenario 2 of a dependency supply-chain attack

Pypi, RubyGems, NPM

- Pypi = Python
- RubyGems = Ruby
- NPM = Node.JS

Results:

Anonymous package creation?	**Yes**
Programmatically Re-package and re-publish?	**Yes**
Run code at install time?	**Yes**

- Ruby have to ‘cheat’ with extensions, but it works

Nuget

- Microsoft, for installing Visual Studio extensions, .NET libraries, PowerShell, Etc.

Results:

Anonymous package creation?	**No**
Programmatically Re-package and re-publish?	**Yes**
Run code at install time?	**Yes**

- Requires a Microsoft Account to upload, which requires a phone number

Apt and Yum

- Ubuntu's/Debian's Apt repos
- Fedora's Yum repos

Results:

Anonymous package creation?	**Yes**
Programmatically Re-package and re-publish?	**No**
Run code at install time?	**Yes**

- Both required real humans to review new packages
- Code review would most likely pick up any dodgyness

Arch User Repository (AUR)

- Arch Linux
- Not main repo, but still heavily use

Results:

Anonymous package creation?	**Yes++**
Programmatically Re-package and re-publish?	**Yes**
Run code at install time?	**Yes**

- Didn’t even verify email!
- Packages are stored in a git repo, so pushing and pulling it the same as Github, etc.

Firefox addons

- Firefox browser

Results:

Anonymous package creation?	**Yes**
Programmatically Re-package and re-publish?	**No**
Run code at install time?	**No**

- Supposedly email address mat be taken into account when assessing your legit-ness
- Submission "may still be subject to further review“
- Implied a human would review the code, but plenty of dodgy addons out there…

Repo	Anon.RBKVTDH package creation	Rapid re-publish	Run code at install time
Pypi	YES	YES	YES
Rubygems	YES	YES	YES
NPM	YES	YES	YES
Nuget	NO	YES	YES
Ubuntu Apt	YES	NO	YES
Fedora Yum	YES	NO	YES
Arch User Repository	YES	YES	YES
Firefox addons	YES	NO	NO

- Basic idea possible with every code repository
- 10minutemail accepted everywhere
- Repos owned by Companies better protected than community
- Attack not very scalable, 1 task = 2 new versions pushed

Mitigations

You:

- Get under the SSL
- Host-based protection
- Understanding of code dependencies
  - Help against supply-chain attack

Public repository owner:

- More protections around who can publish to official repo
- Scanning for packages with constant new releases
  - Both hard to do in an open community

Conclusion

- Trusted service != trusted traffic
- Similar but more advances techniques will exist
- Host-based detection really good idea
  - But can face similar ‘hiding in the noise’ attacks

Questions and thanks

- Vincenzo Iozzo - R.I.P Good Time: https://twitter.com/_snagg/status/10435298376865833O1
- Ruby post-install hook trick: http://blog.costan.us/2008/11/post-install-post-update-scripts-for.html
- This Presentation, other things I've done: https://gitlab.com/users/pathtofile/projects
- Email me for questions, good time: path.to.file[at]gmail