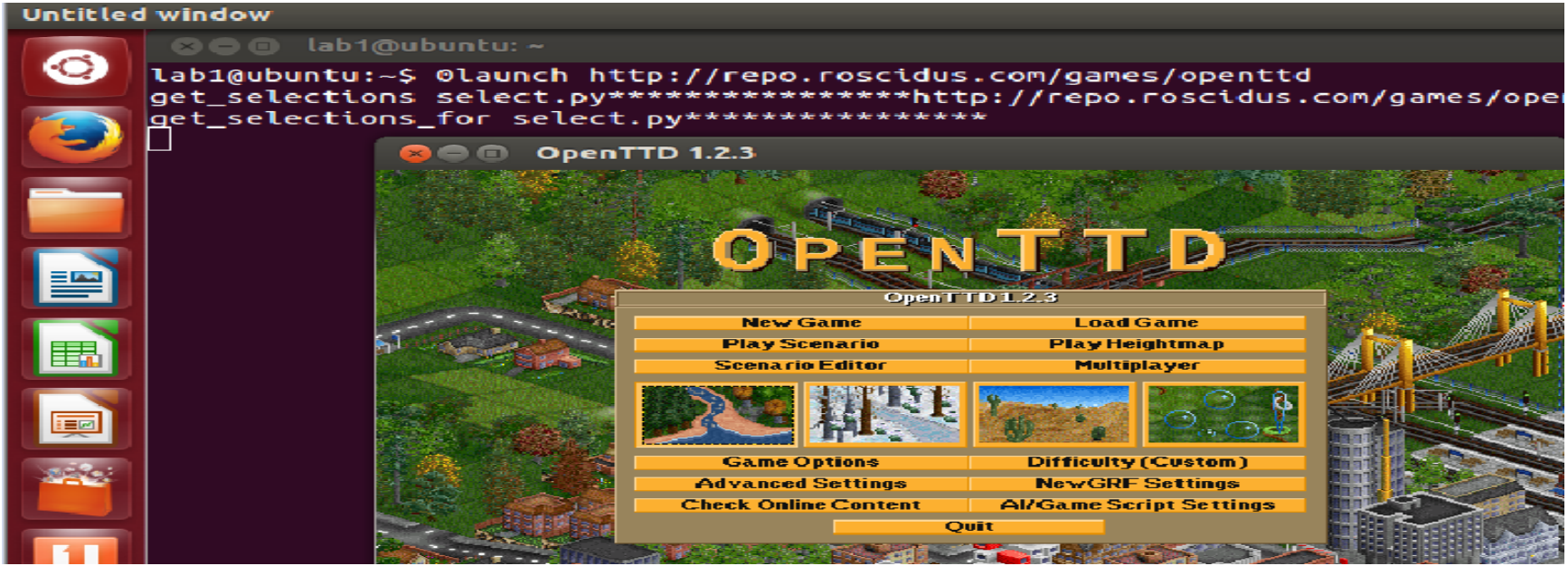
0install + TUF

Team Blue
Alexander Ryzhko
Pawan Ranganatha Rao
Grisha Kumar
Prasit Saebae

What is 0install?

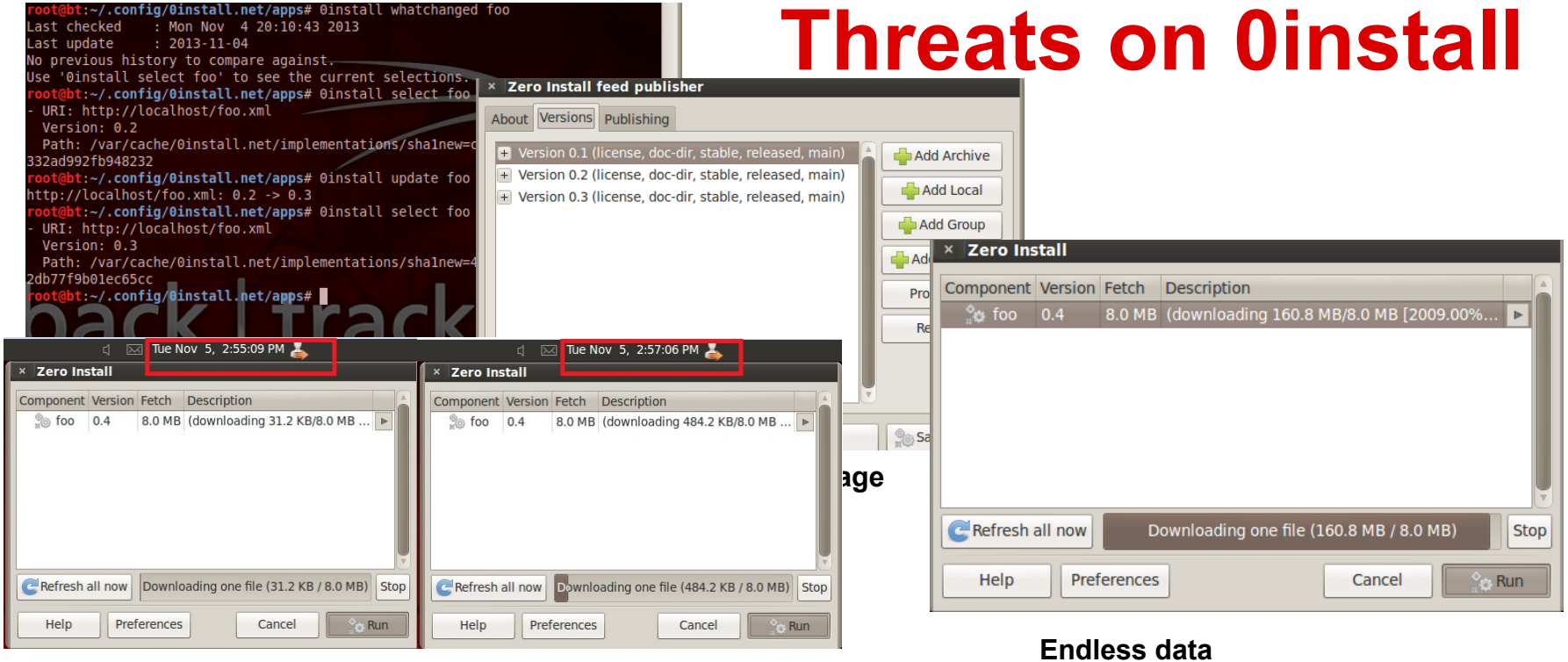
Zero Install is a application that run software without installing it. Packages that are written for 0install can be run on linux and windows OS.



How does 0install work?

- Client enters URI in 0install
- 0install downloads XML feed file
- Through the info in the XML feed file the packages are downloaded
- Client runs the package
- Application runs on the client's system

Threats on 0install

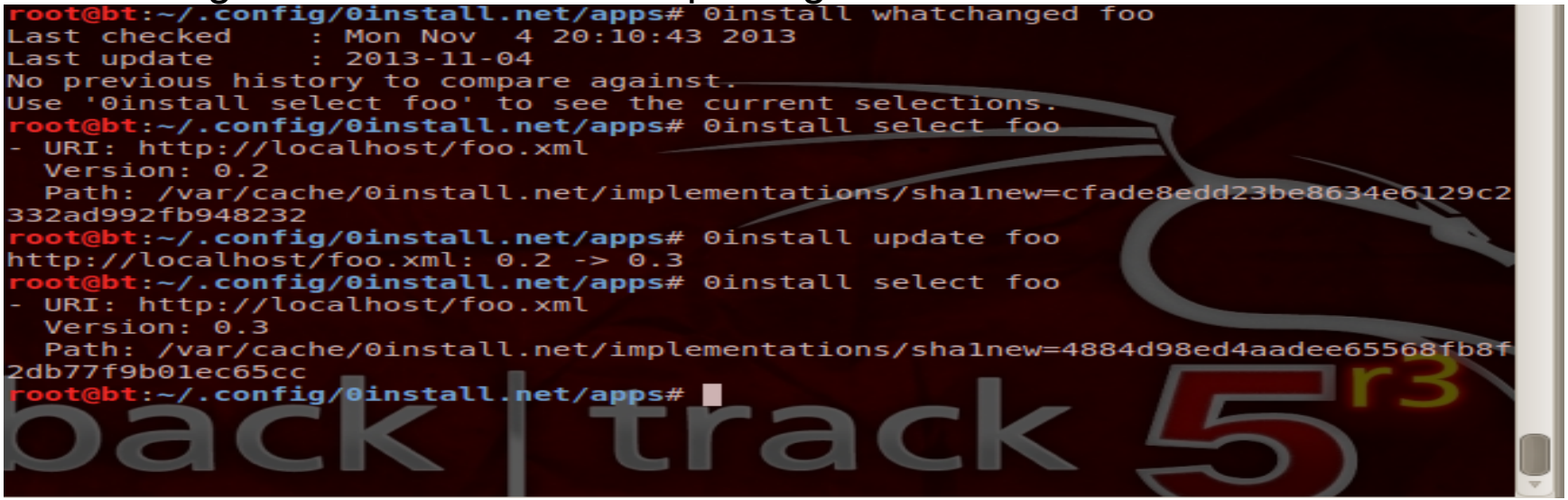


Endless data

Slow retrieval

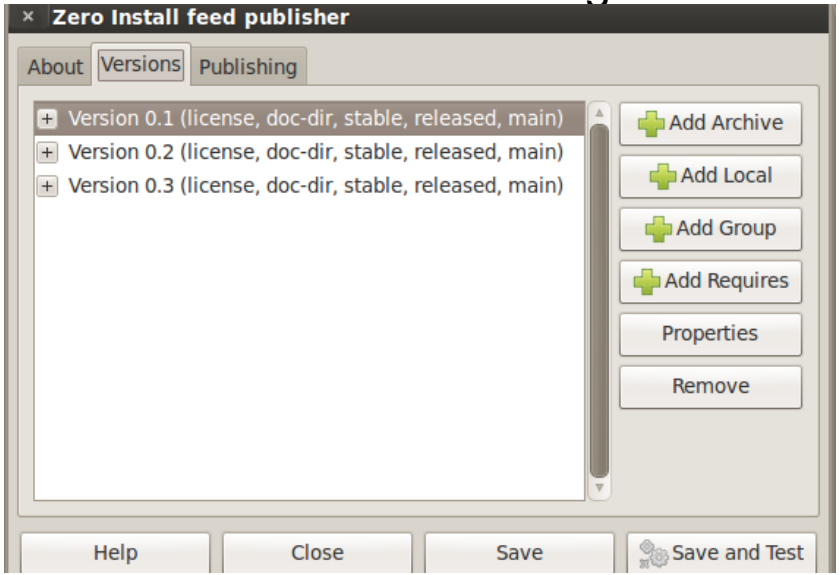
Key Revocation without TUF

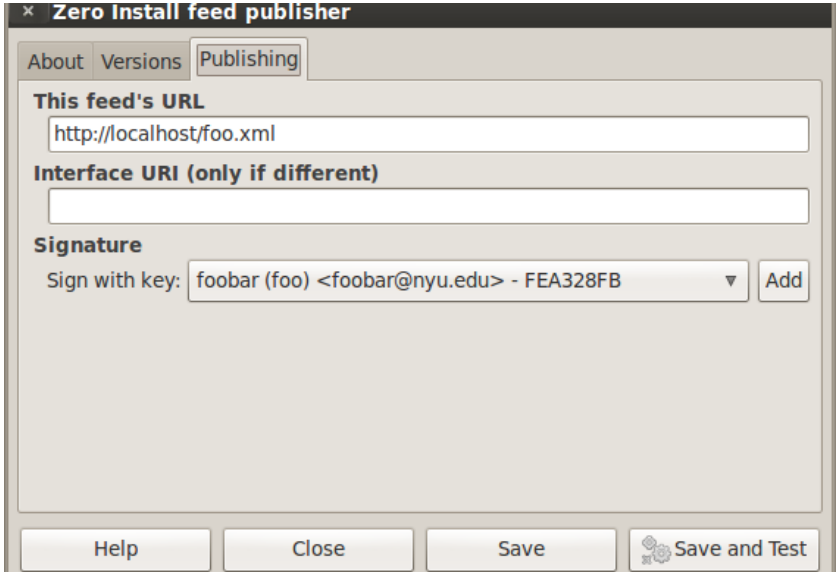
Description: There is a need in key revocation when developer's keys have been compromised. Compromised keys can be used to modify the feed XML file leading to release of malicious package.



Arbitrary packages without TUF

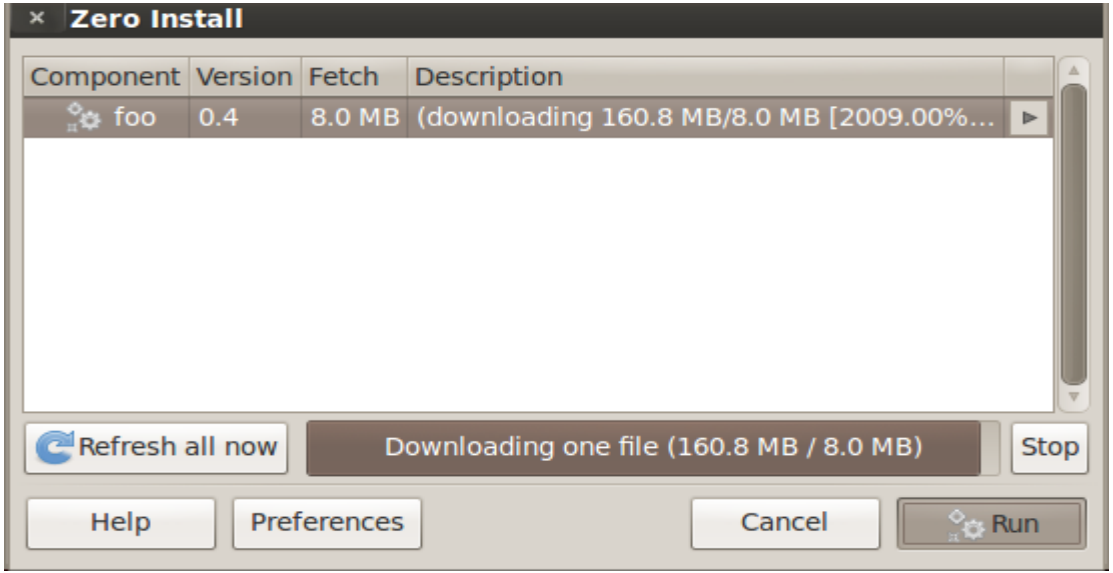
Description: Foo packages with dubious origins. These are malicious packages which an attacker has overwritten over benign packages in order to deceive users into installing them.





Endless data attack without TUF

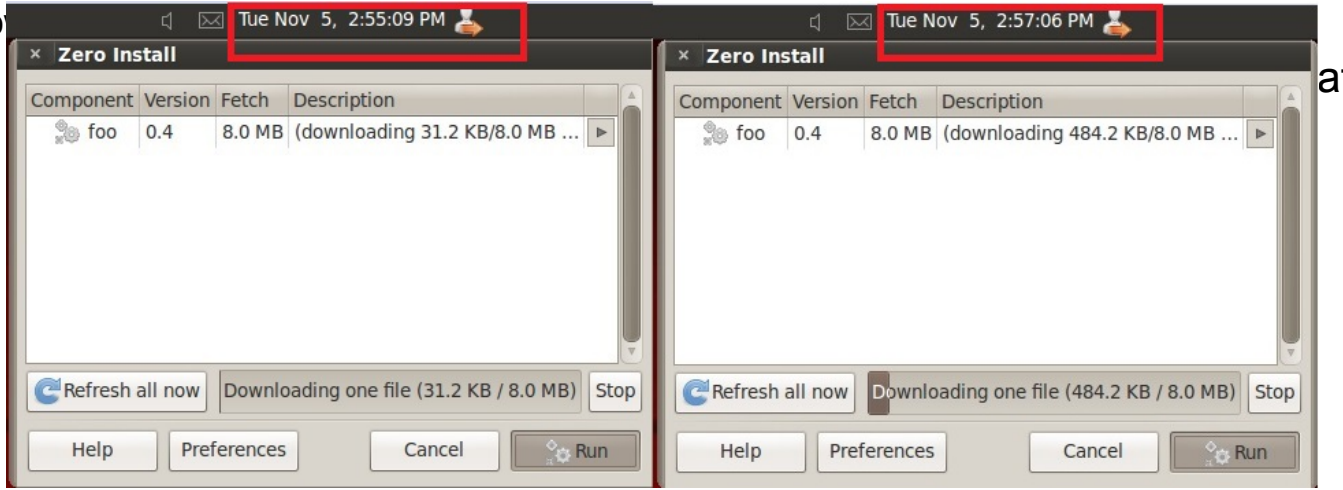
Description: In the process of validating the package 0install downloads the package first. 0install application does not check if downloading file exceeds expected size.



Slow retrieval without TUF

Description: The server sends packets really slowly to the user, the download is really slow and holds up the user. An attacker may deliberately trickle the download in order to remotely exploit a user who is do

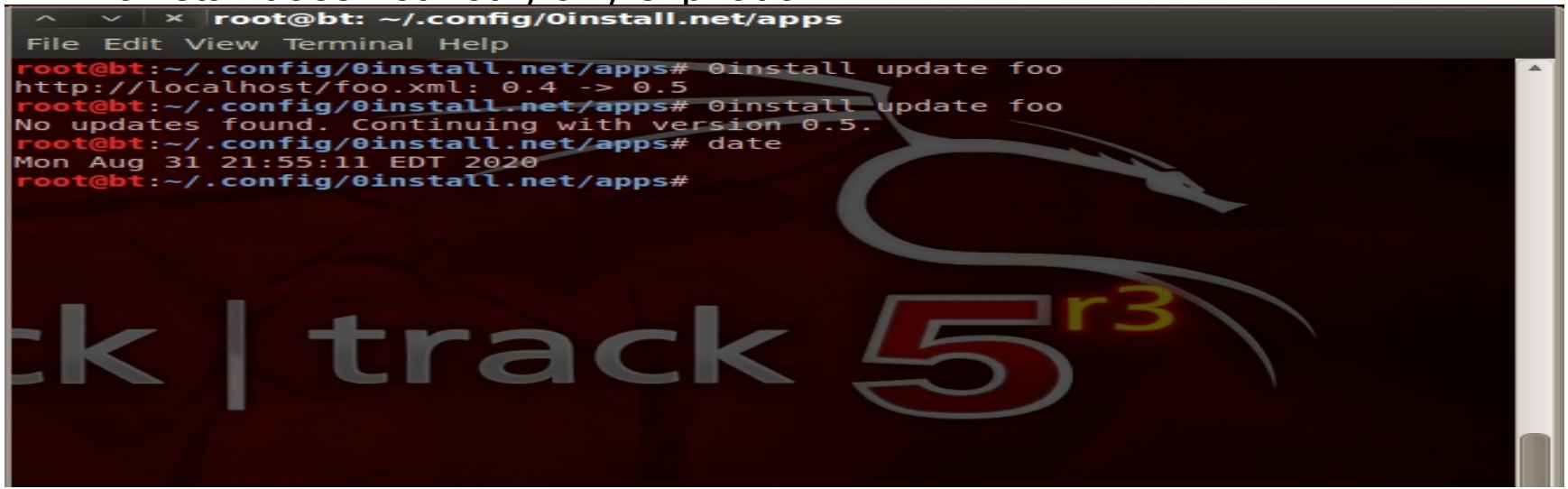
- ate to the user.
-



Freeze Attack without TUF

Description: The freeze attack prevents user from updating possible security updates leaving the user vulnerable.

- 0install does not notify any expiration



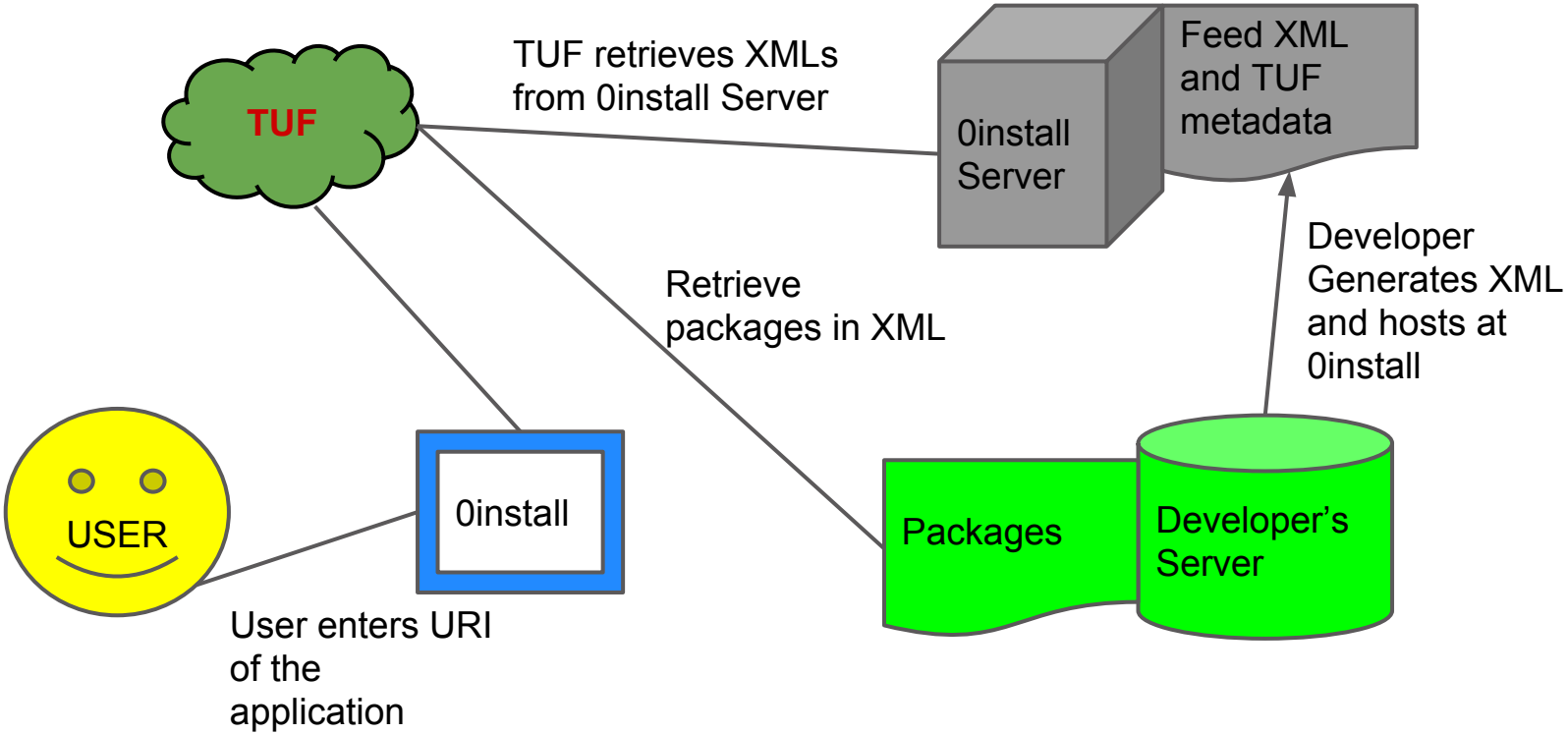
What is TUF?

Software update systems are vulnerable to many known attacks, including those that can result in clients being compromised or crashed.

TUF helps solve this problem by providing a flexible security framework that can be added to software updaters.

We propose to integrate 0install to TUF to fix some of the vulnerabilities discussed above

0install with TUF - Setup



0install with TUF - Implementation

- Both 0install and TUF written in Python
- TUF interposition allows all downloads in 0install to go securely through TUF.

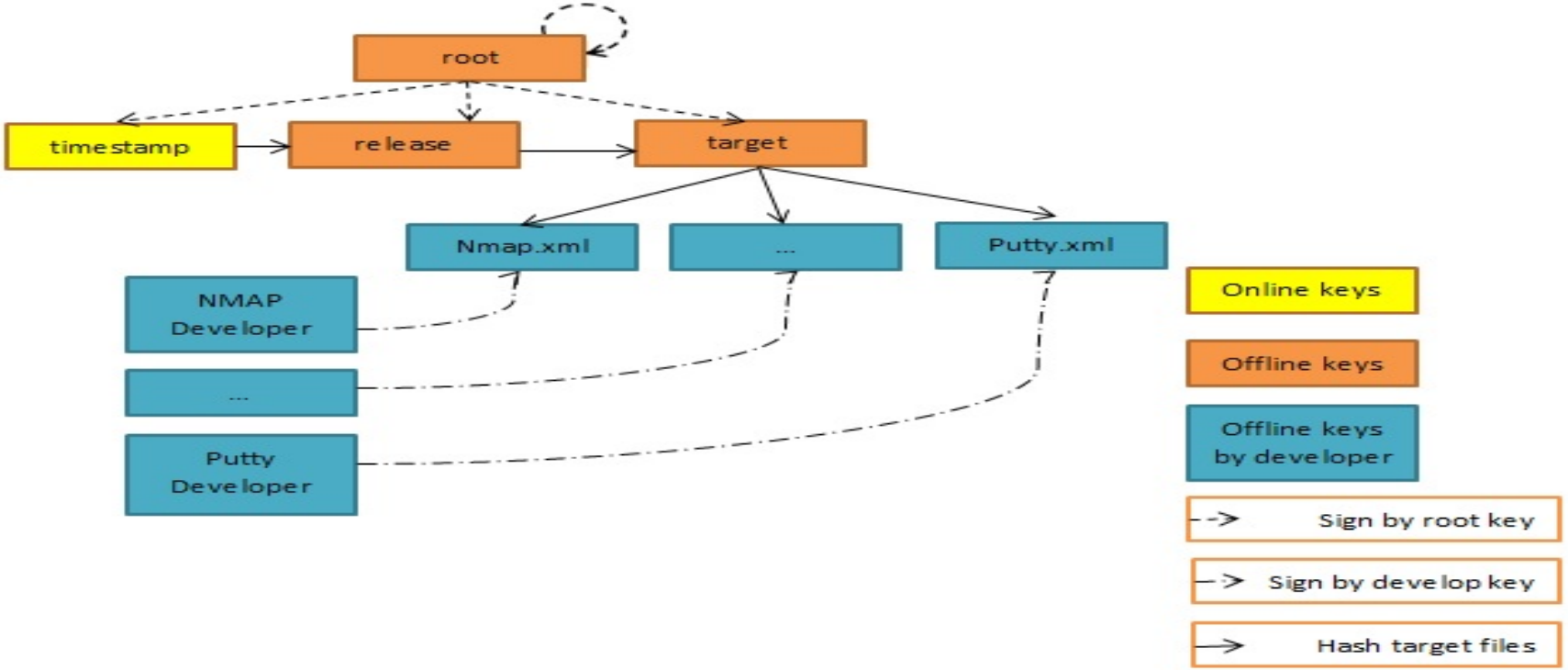
For Feed XML's

TUF metadata is configured on the 0install server for secure download and updated metadata

For Packages

TUF safe download ensures that anything above the expected size is not downloaded

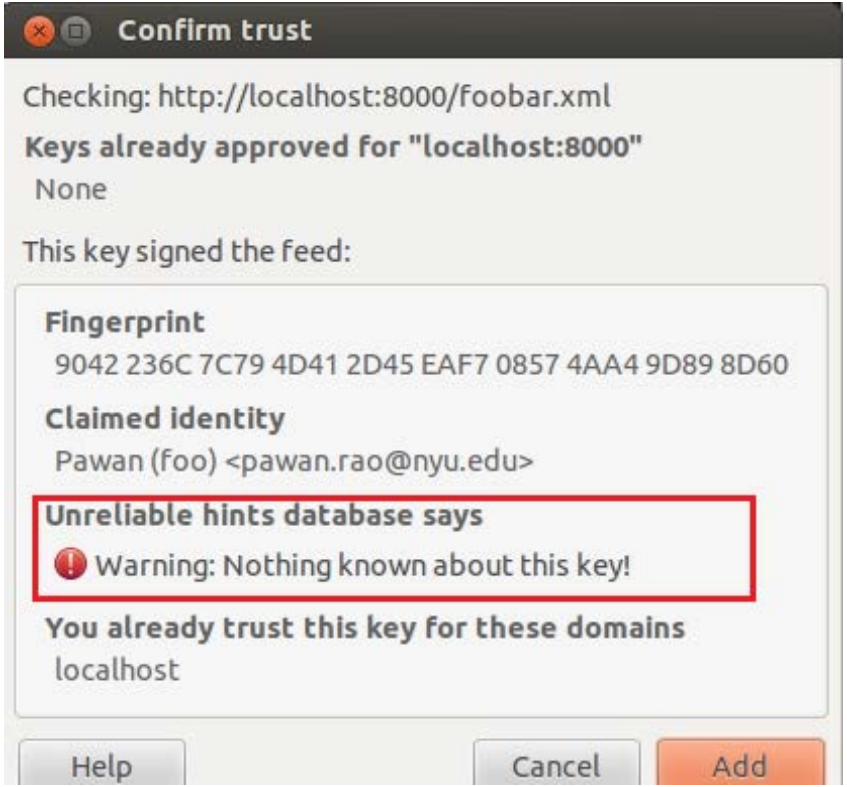
0install with TUF - Metadata & Keys



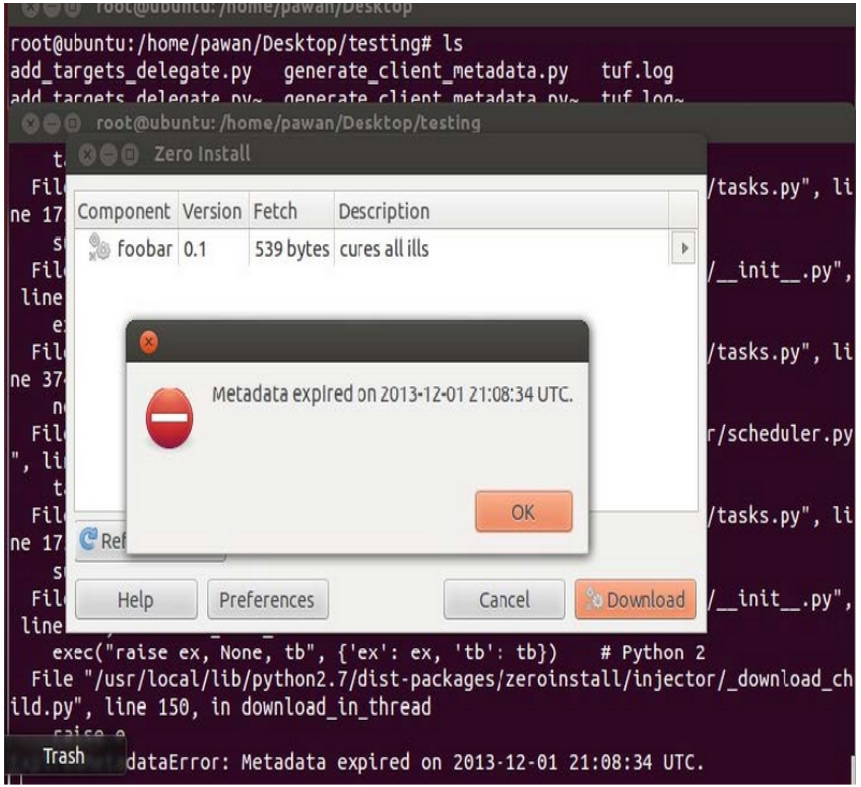
Security

- **Key revocation for developer**
  - Currently, 0install has a dynamic key web service for developer key. We propose that this be a static file which needs to be in the targets file.

- **Arbitrary package attacks - Safe**
  - Since all Feed XMLs are hosted by 0install and are protected with TUF, attacker cannot direct a client to malicious feed XML hosted elsewhere.



Security



- **Slow retrieval attacks - Safe with TUF safe download**
  - 0install with TUF can detect slow retrieval of XMLs only; it will stop downloading and notify user in case server response rate is low.
- **Endless data attacks - Safe**
  - 0install with TUF will download files to amount of file size expected and stop downloading
- **Freeze attack - Safe**
  - TUF protects from freeze attack by raising an error indicating that metadata is expired

Consistency

- How do files get updated?
  - Developer uploads the new xml files to a secure folder on 0install
  - Once update needs to be pushed the metadata is generated and pushed to production from dev environment
- What happens if a client downloads files during an update?
  - We propose that 0install tries to download again after a fixed period of time (e.g. 10 sec), so if the metadata is being updated the user does not get a bad hash error and updates successfully

Efficiency

- Installation of 0install + TUF is simple
- Generating metadata is quick and simple
- Size of the metadata file is 22.31 KB
- User is not affected much with the new implementation

Usability

- User
  - The user does not have to do much with the new implementations of 0install + TUF.
  - The user will have to install 0install and then install TUF on the local system.
  - The rest of the process is more or less the same.
- Developer
  - Register with 0install and claim XML feed file
- 0install admin
  - Generate and host metadata for all XML files

Why use 0install + TUF

There are a few reasons why using 0install + TUF makes more sense than just using 0install

- Safe and simple to use
  - Very small changes to user, developer and 0install
- Uses TUF for updating the packages
  - Everything goes through TUF :)
- 5 keys are used instead of a single developer key
  - Not just one but 5 keys are used so compromise is harder

Why use 0install + TUF

- Installation is simple
  - All packaged together so no additional work to be done
- You know exactly know the source of the package
  - XML and package compromise is harder
- Impersonating developers is harder
  - Offline keys are enforced
- Protects the user from update vulnerabilities
  - TUF mitigates all the update vulnerabilities in 0install

Q & A