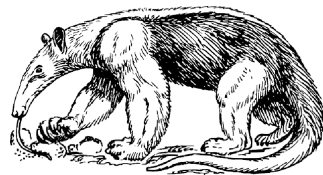# Anteater
CI Gate Security

The Problem

- Approx 1000 Changesets a month / 70+ projects

- No security controls - projects can pull in binaries, scripts, artefacts from anywhere.

- Platform is then deployed into multiple NEP and Operator networks.

Why Anteater..?

Recent Attacks against CI environments

- Ticketing, bug tracking, and git provided secrets (e.g. crypto keys, seeds, hashes, credentials, and source code) provided hacker access to build systems.

- Wikis revealed administrative workflows, IP's and VPN details

- Stolen Engineering credentials (ssh keys) were used to commit backdoors to version control which were self-approved and later deployed into production

https://medium.com/@chrismcnab/alexseys-ttps-1204ef6f2057
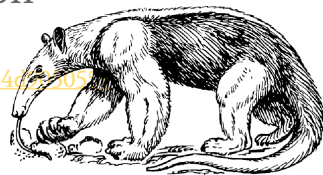
What we have seen in OPNFV..

- Private keys stored in repos (CVE-2016-1000297).
- WGET script downloads from a developer's laptop.
- Lot's of hard coded passwords
- Lot's of binaries
- Uses of 'eval' and other functions that can be exploited.
- Clones of git repositories outside opnfv.
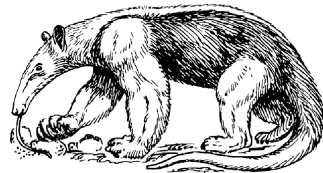
What does Anteater do?

Scans git patches for potential malicious strings or binaries.

If a potential malicious object is identified, it is from merging until reviewed.

How?

Using standard regular expressions to search in scripts / code or any text file:

```
- "-----BEGIN\sRSA\sPRIVATE\sKEY----"
- "curl(.*?)bash"
- "git(.*?)clone"
- "sh(.*?)curl"
- "subprocess(.*?)shell(.*?)=(.*?)True"
```
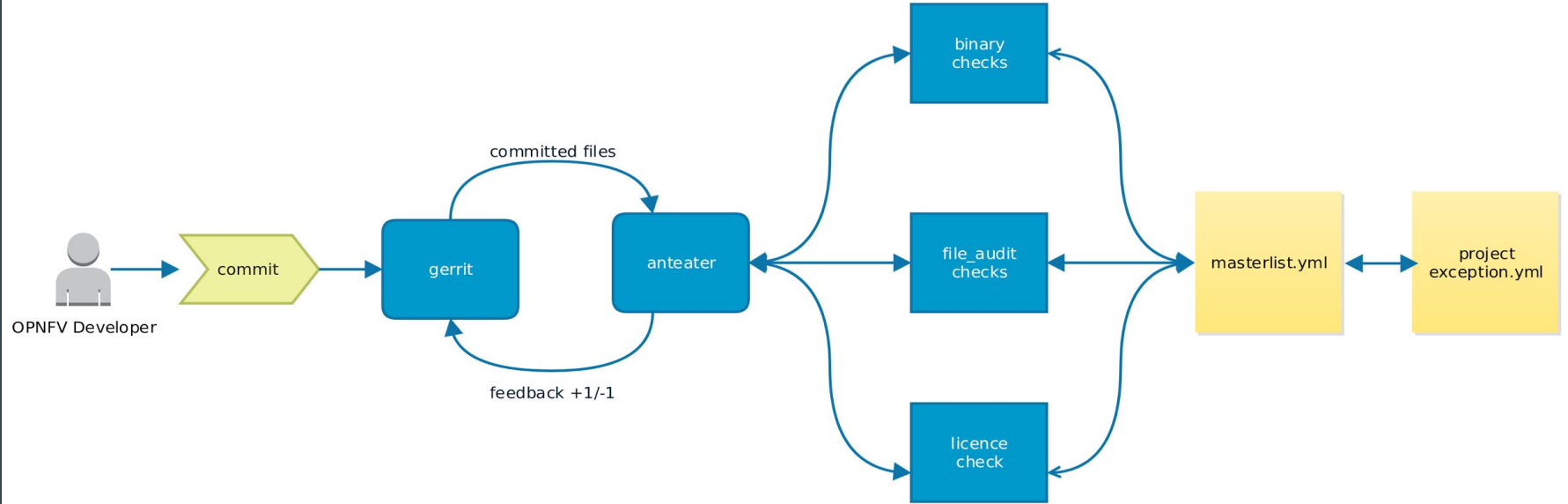
How?

All binary files are blocked, unless a sha256 checksum of the file is provided.

```
[architecture.jpeg, 407fb352a8b709fa1890f200fee5186455fe815fb6c7808305f210e2f1faf76d]
[architecture.pdf, 90517f282ed8137978c9a5e8da06450371fa1a7a783423ee28ba7a5d61f2d1e6]
```

How does this work in CI Gate?

A patchset file list is provided by JJB

```
/home/jjb/repo/fileone.py

/home/jjb/repo/somebinary

/home/jjb/somescript.sh
```

If the file is a binary, its blocked or has an exception (as below).

```
binary_ignore:
  [somebinaryfile, 407fb352a8b709fa1890f200fee5186455fe815fb6c7808305f210e2f1faf76d]
  [anotherbinaryfile, 90517f282ed8137978c9a5e8da06450371fa1a7a783423ee28ba7a5d61f2d1e6]
```

- *Blocked means -1 at gate*

If a file is not a binary, the contents are checked

```
file_contents:
 - -----BEGIN\sRSA\sPRIVATE\sKEY----
 - "curl(.*?)bash"
 - "git(.*?)clone"
 - "sh(.*?)curl"
 - dual_ec_drbg
 - eval
 - gost
 - md[245]
```

Unless an exception is provided:

```
file_contents:
 - "wget http://repo1\\.maven\\.org"
 - paramiko\.RSAKey\.from_private_key_file\(pkey_file\)
 - "git clone(.*)\\.openstack\\.org"
 - "git clone(.*)gerrit\\.opnfv\\.org"
```

Example One

A project needs to clone the following repo in a script / code file:

https://github.com/john_doe/repo

Developer submits a patch to releng-anteater with the following regex:

```
file_contents: "git (.*)github\\.com\\john_doe/repo"
```

Example Two

A project has the following in script

 wget https://trusted.com/somepackage.rpm

Developer submits a patch to releng-anteater with the following regex:

```
someproject:
    file_contents: "wget(.*)trusted\\.com/somepackage.rpm"
```

Example Patch One

Example Patch Two

How is this being phased in.

- During E release, non voting. Voting for F release.

- Tool will be available on PyPi for developers to test locally.

- Works against patchset, not entire repo (although repo can be scanned / daily cron)

- Tools are planned to support developers write and test waivers

New features planned

ClamAV Scanning of all files

HTML rendered reports

Possible Integration with Black Duck Hub API

Developer tools for generating exception regular expressions

Github integration

More information

Wiki: https://wiki.opnfv.org/pages/viewpage.action?pageId=10294496

Git Mirror: https://github.com/opnfv/releng-anteater