Choosing an environment
Authentication and access control
Workflows

Git in the enterprise

Charles Bailey

25th April 2012

Choosing an environment
Authentication and access control
Workflows

1. Choosing an environment
   - Externally hosted
   - Using an "access control" layer
   - Hosting "in house"

2. Authentication and access control
   - Authentication
   - Access control
   - Trusting the content

3. Workflows
   - Overview
   - Scalability
   - Reviewing commits

Choosing an environment
Authentication and access control
Workflows

Externally hosted
Using an "access control" layer
Hosting "in house"

External hosting service

- Less configuration and management
- Reliance on your hosting provider
- Security of your intellectual property

Choosing an environment
Authentication and access control
Workflows

Externally hosted
Using an "access control" layer
Hosting "in house"

Pre-packaged add-ons

- Many to choose from
- Make sure that your choice matches your way of working, or can me made to.
- Possibilities include:
  - gitolite
  - gitosis

Choosing an environment
Authentication and access control
Workflows

Externally hosted
Using an "access control" layer
Hosting "in house"

Just vanilla Git

Total control... but everything extra that you want must be
scripted yourself.

Consider development and maintenance costs.

Choosing an environment
Authentication and access control
Workflows

Externally hosted
Using an "access control" layer
Hosting "in house"

Hosting it yourself

Vanilla `git-web` will only get you so far

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

Authentication through transport

- Git has no native authentication
- ssh is the transport of choice for strong authentication
- Only use git transport in read-only mode

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

Trust and keys

- Using passwords with Git+ssh is laborious
- Public key authentication is easier
- The ability to revoke keys is essential

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

Account configuration

- Universal git account with multiple keys
- Individual accounts

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

"Commit" access

Historically in Open Source, commit access was for the privileged.
In the commercial environment, direct access to save to the
repository is a requirement to "get the job done".

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

ssh access

- Direct write access means no controls
- `git update-ref -d refs/tags/v1.0` is dangerous
- ... but then so is `rm -rf repo.git`

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

Safer options

Levels of access

1. Direct ssh access to central server
2. Commands restricted by ssh key restrictions
3. Restricted git-shell access

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

Finer grain

- File system permissions can give per-repository read and write access
- Per-branch control is usually achieved through the update hook E.g. `contrib/hooks/update-paranoid`

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

Recommended configuration

```
git config receive.denyDeletes true
```

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

Auditing

Pusher vs author vs committer

- Enforce known "author" — but anyone can spoof another's commit and this disallows pushing others' commits in rebases and new branches
- Enforce known "committer" — still easy to forge
- Enforce "committer = pusher" (e.g. `update-paranoid`) — but now I can only push another's commit if I amend it
- Audit pushes

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

Trusting the content

- Signed tags
- Signed commits

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

Signed commits

- Requires Git $>= 1.7.9$
- Signature is stripped on *any* modification of the commit
- A signed commit guarantees the signer (not the committer or author)

Choosing an environment
Authentication and access control
Workflows

Authentication
Access control
Trusting the content

Signed commit demonstration

Choosing an environment
Authentication and access control
Workflows

Overview
Scalability
Reviewing commits

Yes, Git supports all your workflows

- Git gives you great freedom.
- You may have to script your own enforcement.

Choosing an environment
Authentication and access control
Workflows

Overview
Scalability
Reviewing commits

Free-for-all

- Don't expect the ultimate in cleanliness
- Education is important
- Beware the "cheat sheet"

Choosing an environment
Authentication and access control
Workflows

Overview
Scalability
Reviewing commits

Locked down

- Slows down your development rate
- *May* increase quality
- *May* make for a cleaner history
- *May* backfire

Choosing an environment
Authentication and access control
Workflows

Overview
Scalability
Reviewing commits

Forms of lock-down

- "Golden" repository has restricted access
- Release branches have restricted access
- Commits need appropriate "signature"

Choosing an environment
Authentication and access control
Workflows

Overview
Scalability
Reviewing commits

Large objects

- Use Git for source
- Avoid placing binaries and large objects in your source tree

Choosing an environment
Authentication and access control
Workflows

Overview
Scalability
Reviewing commits

Pushes per day

Total push frequency is a constraint

1. Team size
2. Per developer push frequency
3. Test cycle length

Choosing an environment
Authentication and access control
Workflows

Overview
Scalability
Reviewing commits

Partitioning the world into repositories

- Git's "whole tree" versioning makes partitioning important.
- You can't get away with a "universe" repository.
- Put the source for each unit of release in to a single repository.

Choosing an environment
Authentication and access control
Workflows

Overview
Scalability
Reviewing commits

Submodules

To reflect a submodule change, a commit *must* be made in the parent repository.
This means that submodules don't solve the "contended HEAD" problem.

Choosing an environment
Authentication and access control
Workflows

Overview
Scalability
Reviewing commits

Reviews

- `Signed-off-by:` can mean what you need it to mean
- Review commits before pushing
- Allow reworks to be squashed in
- Decide whether to review merges

Choosing an environment
Authentication and access control
Workflows

Overview
Scalability
Reviewing commits

Questions

Questions?