

Submitting Patches

Contributing to the IPFire project made easy

Michael Tremer

IPFire Project

October 10th, 2015

We are an Open Source project

We are an Open Source project
... and use an open development approach

Everyone is welcome to contribute

No skills required

This might look complicated
- it is mostly common sense

Why does all this matter?

Why does all this matter?
Our brains are good with patterns

We are using `git`

Everything we do is in a git repository somewhere on
`http://git.ipfire.org`

Setting up Git

Introduce yourself

We want to know who you are
Real name required

Setting up Git

Introduce yourself

We want to know who you are
Real name required

```
git config --global user.name "Michael Tremer"  
git config --global user.email michael.tremer@ipfire.org
```

Setting up Git for Email

Git needs to send email...

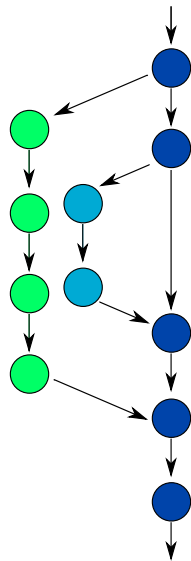
```
git config --global sendemail.smtpencryption tls
git config --global sendemail.smtpserver mail01.ipfire.org
git config --global sendemail.smtpserverport 587
git config --global sendemail.smtpuser ms
```

(You can use what ever mail server you want...)

Creating a Development Branch

We use development branches for each feature

```
git checkout -b my-feature -t origin/master
```



Time to hack...

Time to hack...

But there are some rules...

Check with us before starting working on bigger things

Check with us before starting working on bigger things

Separate your changes
e.g. bug fixes vs. performance enhancements

Check with us before starting working on bigger things

Separate your changes
e.g. bug fixes vs. performance enhancements

Send small patches
each one justifiable on its own merits

Check with us before starting working on bigger things

Separate your changes
e.g. bug fixes vs. performance enhancements

Send small patches
each one justifiable on its own merits

Make sure that everything builds
mark dependencies

Check with us before starting working on bigger things

Separate your changes
e.g. bug fixes vs. performance enhancements

Send small patches
each one justifiable on its own merits

Make sure that everything builds
mark dependencies

Style-check your changes

Check with us before starting working on bigger things

Separate your changes
e.g. bug fixes vs. performance enhancements

Send small patches
each one justifiable on its own merits

Make sure that everything builds
mark dependencies

Style-check your changes

Test!

Committing Changes

Committing a patch means
adding it to the project's history

Committing Changes

Committing a patch means
adding it to the project's history

It will be in there for forever

Writing a good commit message

Describe your changes

Writing a good commit message

Describe your changes

What motivated you to develop this patch?

What is the impact of the patch?

Are there user-visible changes?

Writing a good commit message

Describe how you solved the problem

Writing a good commit message

Describe how you solved the problem

Include technical details

(If this step is getting difficult your patch is probably too complex)

Writing a good commit message

Add useful information

Writing a good commit message

Add useful information

Bug IDs

Mailing list discussions
(including a quick summary)

Git tags - Who did what?

Sign your work

```
Signed-off-by: Michael Tremer <michael.tremer@ipfire.org>
```

Git tags - Who did what?

Credit your collaborators

Reported-by: | Tested-by: | Reviewed-by: | Suggested-by:

Why all this?

Why all this?

Commit messages will be read by the reviewers of your patch...

Why all this?

Commit messages will be read by the reviewers of your patch...
(help them to make reviewing your changes easier)

Why all this?

Commit messages will be read by the reviewers of your patch...
(help them to make reviewing your changes easier)

...and after many years

Every patch must be reviewed

Respond to any comments - don't get discouraged - or impatient

Who to send patches to?

The maintainer of a subsystem

and at least one mailing list

We use email only

Pull requests only for bigger changes & maintainers
No GitHub pull requests

Patch format

- Must include a From: header (with the real name and email address)
- Subject line starting with [PATCH]
- The commit message
- Signed-off-by: line
- The actual changes in the unified diff format

Patch format

- Must include a From: header (with the real name and email address)
- Subject line starting with [PATCH]
- The commit message
- Signed-off-by: line
- The actual changes in the unified diff format

`git send-email` automates all that

Track what is going on...

`http://patchwork.ipfire.org/`

Track what is going on...

`http://patchwork.ipfire.org/`

Help us reviewing patches



PC Engines

Lightning Wire Labs

We bring future technology into your network



Read everything on

<http://wiki.ipfire.org/devel/submit-patches>