

Open-Source and the corporate world

Ivan Nardi

Who am I?

- Ivan Nardi, @ AI2M:
 - lawful interception, investigation analysis, big data retention
 - voice/IP metadata collection, processing and reporting
 - network probes and DPI
- ivan@ai2m.eu

Agenda

Businesses and Open-Source: why (not) giving back to the community?

A lot of companies use, modify and integrate open source software. Should they push to upstream their changes?

Why not?

- The employer doesn't want it
- The patch contains proprietary/confidential information
- The patch give you an advantage over your competitors
- Getting your patch accepted is hard for a variety of reasons:
 - You need to follow some contribution guidelines: coding convention; good commit message; anonymous attachments
 - You need to draw the attention of the project maintainer(s)
 - Your patch might be simply wrong or “uninteresting”
 - Your patch might need some modifications: Continuous Integration error; performance regression; multi architecture support; a more general solution

Why yes?

- To give back to the community
- It is a kind of advertising
- Before merging, your patch could be improved
 - You might get some reviews (by skilled people)
 - You might get some more CI tests and fuzzing
- After merging:
 - Your patch will be automatically maintained
 - You can forget it; when you download a newer version, you will get your modification, too

Managing a fork

Does all of that really matter? Probably not...

However, if:

- Your patches amount/complexity increase over time
- You frequently pull from upstream

You end up managing a private/personal (diverging) **fork**

It requires a lot of **time** (tests, rebase/merge,...)

If something breaks, how do you find the culprit?

- Upstream changes? Your local patches? Some conflicts wrongly resolved?

Conclusions

Pushing to upstream your changes might save you a lot of time