written by Thomas Maroulis on 12 January 2017 in devops, engineering

security, secure by default

This week we are continuing the security series of posts by introducing another concept. That of being secure by default. This idea can be thought of as an attempt to work around the biggest flaw in every security system ever built: human error.

It's a fact of life that there is no system, no matter how well engineered it is, that is not susceptible to human error. Expecting that procedures will always be followed to the letter is setting yourself up for failure when inevitably someone somewhere slips up. There are of course ways to deal with that. Security audits is one popular such mechanism, but it can only find errors after the fact. Being secure by default is another mechanism that can complement audits by trying to prevent errors in the first place.

secure by default

The idea behind secure by default is a simple one. Inaction should always result in a system/person/etc having fewer or equal privileges compared to what they need.

To offer an example, if a new service needs to talk to some other services to work properly then a good default could be for a new service to have no privileges at all. If proper configuration of this service is forgotten then the service will not work. That should hopefully be noticed by those trying to use it, the problem will be identified and fixed. Conversely a bad default would be for the service to have all available privileges and expect someone to trim them down after the fact. If a mistake were to happen in the second case the service could very likely run for a long time with privileges it should not have until an audit detects the mistake.

This operates under the principle that people are often more likely to forget something than to do it wrong once they realise it has to be done.

Another way to phrase this in a more general setting would be this. Configure your systems so that the result of a mistake of omission is a visible error that will attract attention to itself rather than a silent error that might go unnoticed.

That's all from me. See you again in two weeks.

past posts in the series

- mind your entry points
- soft shell no more

If you enjoyed the read, drop us a comment below or share the article, follow us on Twitter or subscribe to our #MetaBeers newsletter. Before you go, grab a PDF of the article, and let us know if it's time we worked together.