Close the Gap: Bringing Engineering & Security/Compliance Together

Sare Ehmann

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM20-0902

Agenda

Sources of Friction

Example: Little Fish & Tech Giant

Repairing the Damage

DEV SEC OPS DAYS

Sources of Friction

Developers

- Holding me back!
- Interrupts automated DevOps processes
- Takes *forever*
- Unexpected failure
  - But it passed last time and we didn't touch that part!
- Don't know what they want, can be intimidating

Security

- Always needed *yesterday*
- Don't trust developers
- Sisyphean effort, Atlassian responsibility
- Personal responsibility/liability
- New tech stacks, new ops stacks, world changes quickly under their feet

DEV SEC OPS DAYS

Example: Little Fish & Tech Giant

The story of Little Fish

- ~20 developers; ~4 years in business
- Bought by Tech Giant
- Scaled to ~100 developers in 6 months
- Old talent left, new talent under-trained
- Tech Giant audit found vulnerabilities
  - Tech Giant doesn't trust Little Fish developers – they're cowboys!
  - Tech Giant enforces security audit on all new code
    - Enter the problem…

Little Fish Developers & Tech Giant InfoSec

- Both resentful
  - Tech Giant InfoSec likes waterfall, Little Fish is agile
  - Tech Giant InfoSec likes documentation, Little Fish has been too small for that
  - Tech Giant InfoSec wants completed product, Little Fish works in MVP and JIT
  - Tech Giant InfoSec waaaaay overworked, tasked with ‘fixing’ Little Fish
- Horrible communication style
  - Aggression to face
  - Passive-aggression behind back
  - Throwing each other under leadership eye
  - Shouting in meetings

How did we fix it?

DEV SEC OPS DAYS

Repairing the Damage

Repairing the Rift

1. Change communication dynamics
2. Changed paradigm for interaction with InfoSec
3. Produced artifacts that supported the new interaction
4. Created living artifacts

Communication Dynamics

1. Created subteam to handle all InfoSec communications until things were improved
2. Patience & empathy out the ears
3. Reiterate: want to make your job simpler
4. Tighten feedback loop – shorter, faster communication

Interaction Paradigm

1. Moved to ”help me decide” vs. “stamp of approval”
2. Not easy

New Artifacts

1. Stopped using old templates – gave wrong impression
2. Asked what they wanted, delivered and iterated on that until satisfied
3. Always added/modified, never removed

Living Artifacts

1. Used gitbook to create on-the-fly documentation packets for InfoSec as needed
   - Used Mermaid JS to create charts in Markdown
2. Deployed the gitbook in pipeline for InfoSec usage
3. Used git SHAs on sign-off from InfoSec to show good faith and create audit trail
4. Updating markdown files now part of developer ettiquitte

Was it Successful?

DEV SEC OPS DAYS

Recap

Agenda

Sources of Friction

Example: Little Fish & Tech Giant

Repairing the Damage