Locking Them out of Their Own House

Jackie Bow
Patreon
My Past Life
The Startup Tech Stack

- Maybe upwards of 150 people
- 2 - 5 years
- Caught your stride in product offering, bright future
The Startup Tech Stack

- Undocumented
- Ambiguous ownership
- Full of technical ghosts and gremlins
- It “works”
- Culturally weighted
That Seems Terrible.
That Seems Terrible.

- It’s actually the sweet spot.
  - Form a culture with security as base value
  - Foster Security champions across organizations
  - Implement security that really makes lives easier
    - Endear yourselves before you have to take things away
Example: Manage Access to Cloud

● “We need you to figure out access control to our cloud resources”

Or, more likely:

● “We’re not sure how we manage access to our cloud things”
Example: Manage Access to Cloud

- Goals:
  - Enable people to do their jobs
  - Secure access to critical resources
  - Do so in a way that others can do after you
Example: Manage Access to Cloud

- How? Make it easy.
  - For employees
    - To access (and request access)
  - For IT/Security
    - To grant/revoke permissions
    - To update permissions
    - To monitor what is happening
Example: Manage Access to Cloud

- Leverage what you have.
- Make it manageable.*

Then,

1. Establish source of truth
2. Triage the patient
3. Determine needs
4. Implement solution
5. Iterate
Step 1: Source of all Truth

- What is your source of truth for employees?
  - LDAP
  - HRIS
  - Okta/Duo

- Centralized identity management early = less headaches later.
Step 2: Triage the Patient

- **Structure**
  - 1 account or many?

- **Access Methods**
  - IAM users
  - IAM roles

- **Pruning**
  - Your Founder does not need Admin AWS access.
Step 3: Determine Needs

- How people currently interact / access?
  - Console
  - CLI
  - Hybrid
- How should* they?

* Again: Avoid trying to make the most “secure” or beautiful plan. Humans are messy and will do the easiest path.
Step 4: Implement: SSO/SAML

- SAML is easy* to set up
- NO users or keys to manage
- Management is centralized
- AWS access not needed to manage
Step 4: Implement: Setting Roles

- Access granted through Assumed Roles
  - Create + Update in code
  - Don’t overscope roles to begin
  - Create general buckets
    - Job function? Purpose?
  - Use Naming conventions
Step 4: Implement: Create Process

- Documentation! Roles and their uses
- Requesting access
- Break Glass method for emergencies
Step 4: Implement: Monitor.

- Monitoring
  - Cloudtrail
  - Custom SIEM/Logger
- Dashboards or tracking
  - Critical to understand use going forward
Step 5: Roll out!

- Roll out incrementally
- Communicate user removal directly
- Focus on communication of ease-of-use
Step n+1: Iterate

- This step is never done.
- There will always be updates that need to happen
  - New tools
  - Re-orgs
  - Special cases*

* There will always be a few engineers who believe they need access to everything. Pick your battles.
TL;DR

- Security works best when it serves the users
- Iteration and manageability are key in the startup environment
- Never ignore emotion as it relates to tech ownership and history
Questions?

- jbow@patreon.com