

The deputy shot the sheriff



Privilege escalation in build pipelines

Andreas Sieferlinger

Senior Cloud Platform Engineer

@webratz



it's not just your employees

JavaScript Packages Caught Stealing Environment Variables

By [Catalin Cimpanu](#)

August 4, 2017 08:42 AM 2



On August 1, npm Inc. — the company that runs the biggest JavaScript package repository — removed 38 JavaScript npm packages that were caught stealing environment variables from infected projects.

Security

Check your repos... Crypto-coin-stealing code sneaks into fairly popular NPM lib (2m downloads per week)

Node.js package tried to plunder Bitcoin wallets

By [Thomas Claburn](#) in [San Francisco](#) 26 Nov 2018 at 20:58 49 SHARE ▼

RESEARCHERS FOUND BACKDOOR IN PYTHON LIBRARY THAT STEAL SSH CREDENTIALS

Share this...



Recently we saw an attempt to hide a back door in a code library, and today there is a new case. This time, [information security](#) experts found the backdoor in a Python module.

In the SSH Decorator module (ssh-decorate), created by the Israeli developer Uri Goren, which is a library for handling SSH connections from the Python code.

Agenda



- evolution of software delivery systems
- potential problems
- solution example
- mitigation strategies

q & a

Andreas Sieferlinger

Senior Cloud Platform Engineer

SCOUT 24

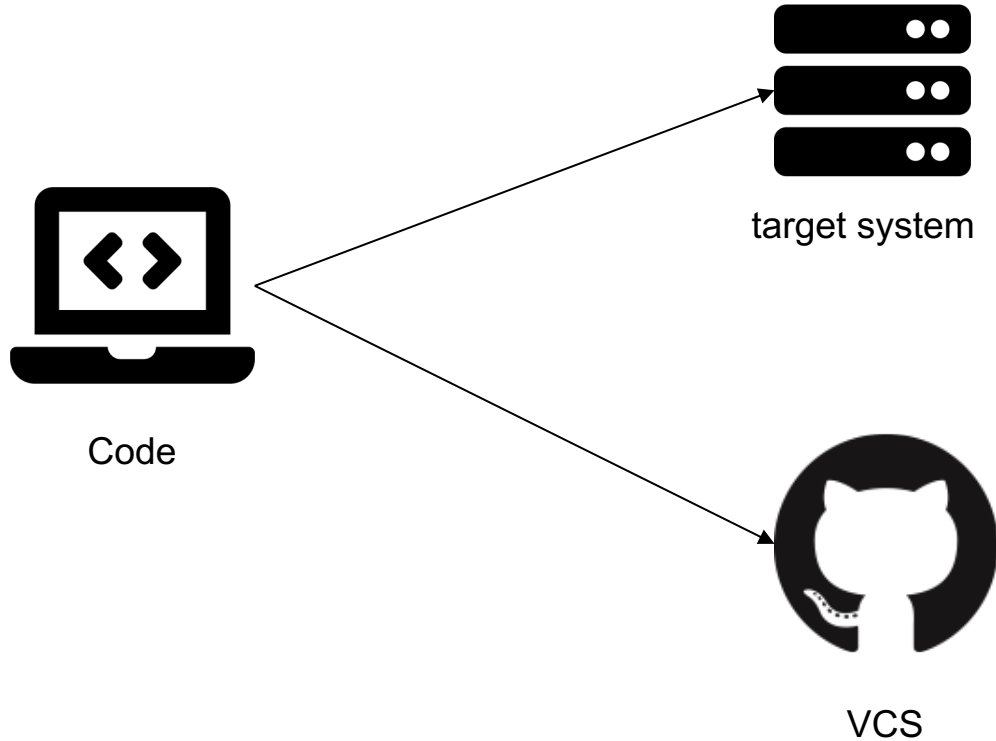
Twitter: @webratz

All things AWS, CI/CD,
GitHub

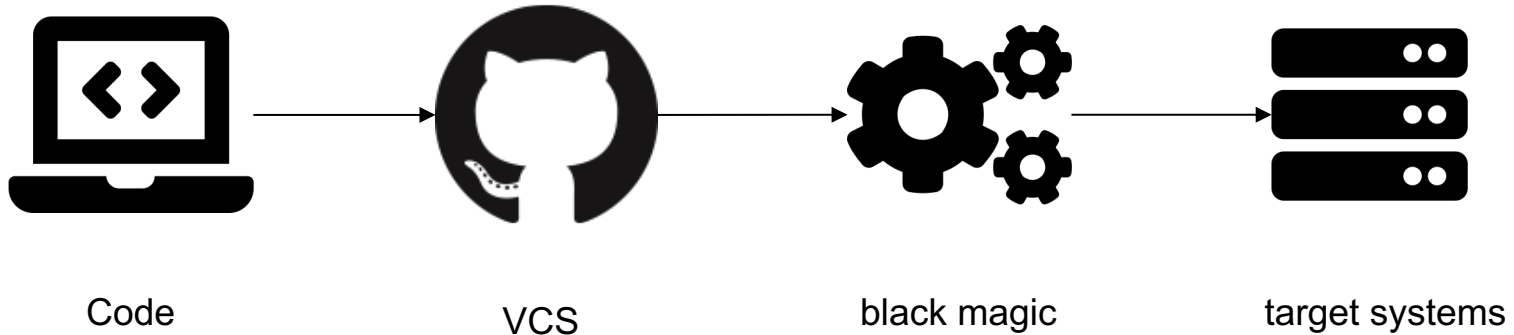
Evolution of Software Delivery



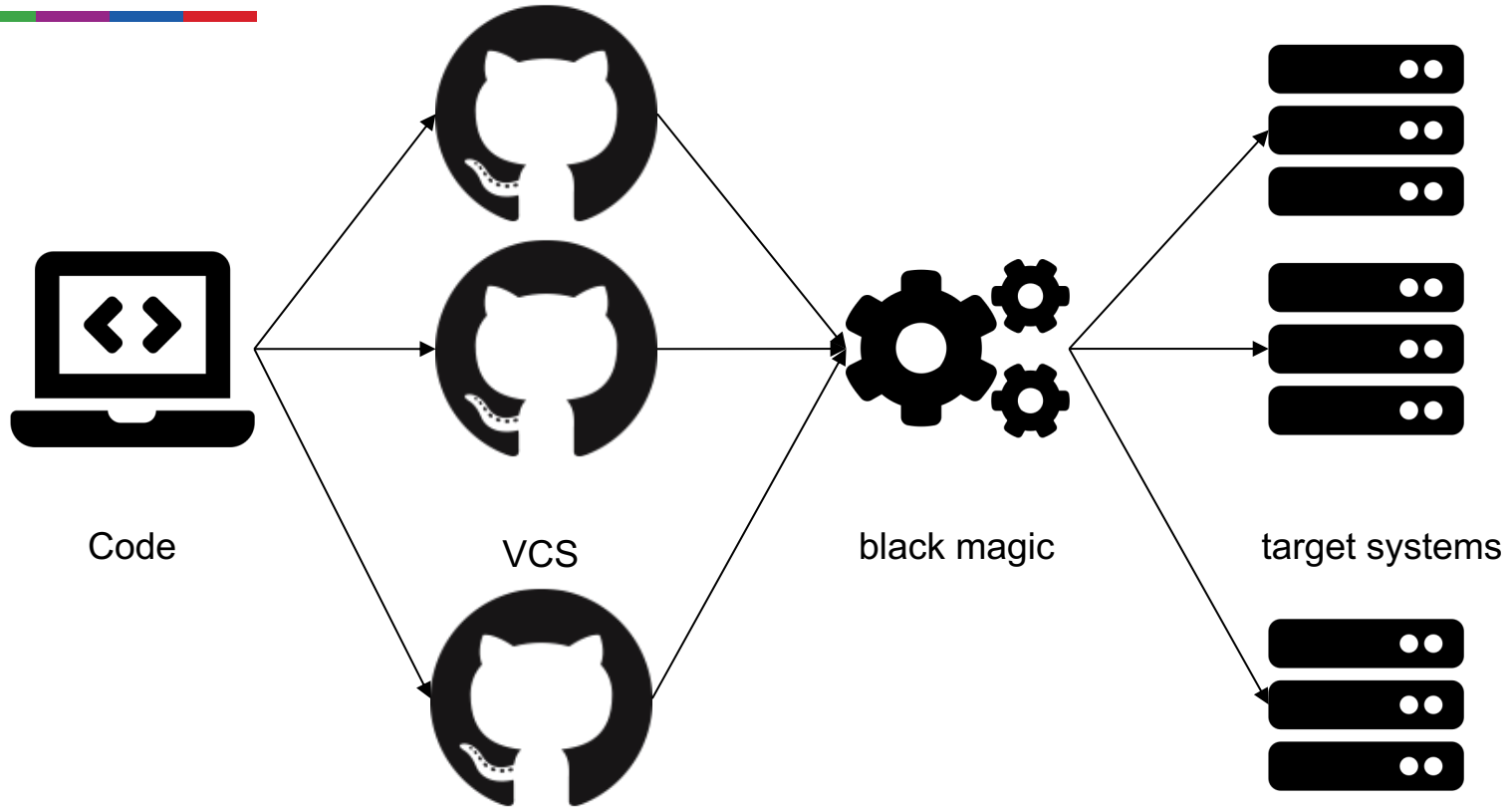
Evolution of Software Delivery



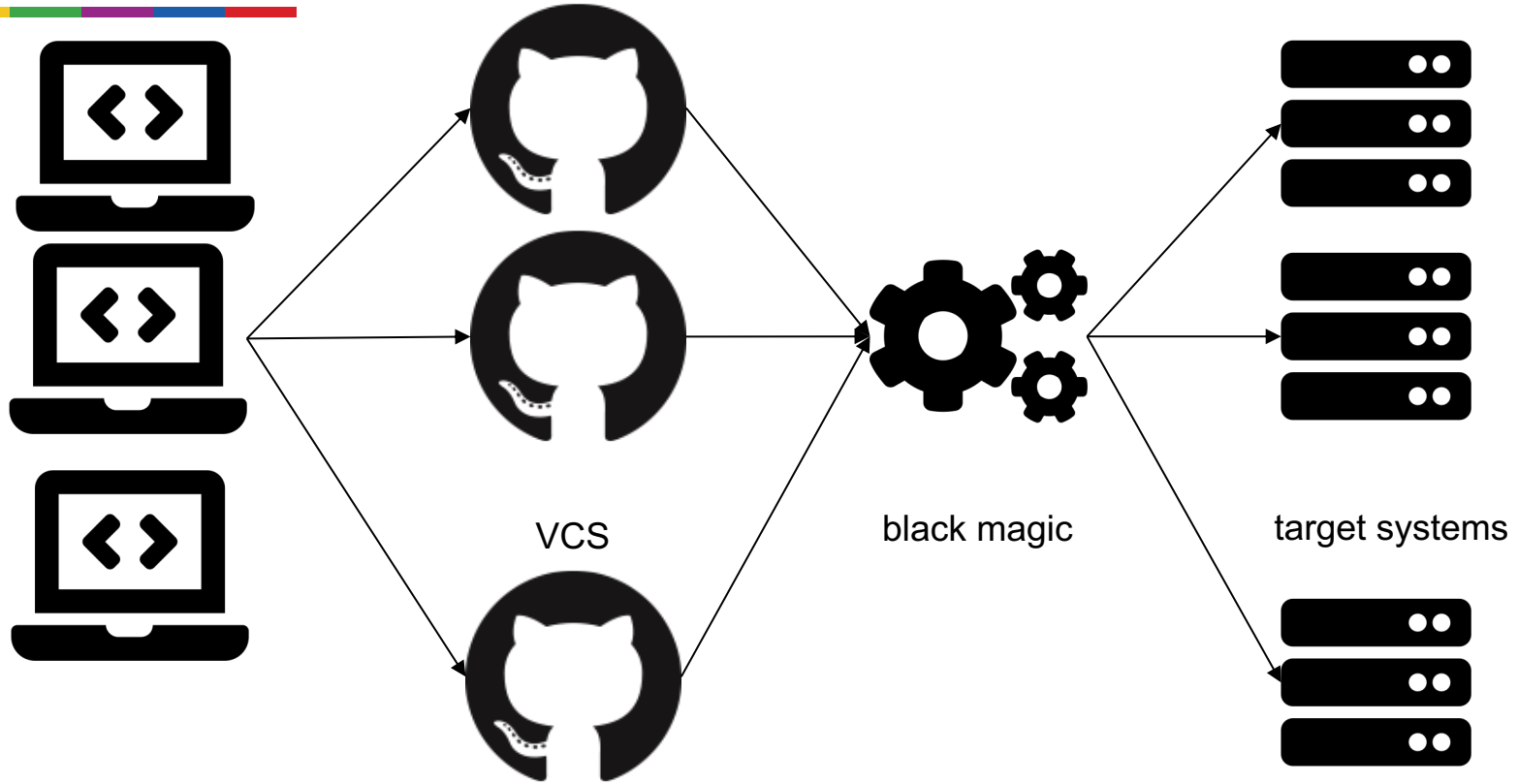
Evolution of Software Delivery



Evolution of Software Delivery



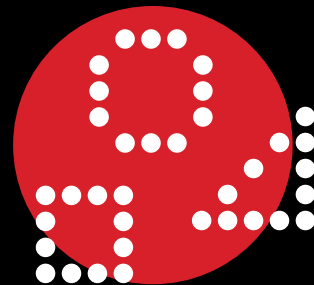
Evolution of Software Delivery



Evolution of Software Delivery



- Workflow got more complicated
- More involved components
- often shared components
- bigger user base
- often very centralized



black magic



it's not

What does a CI/CD pipeline



Code

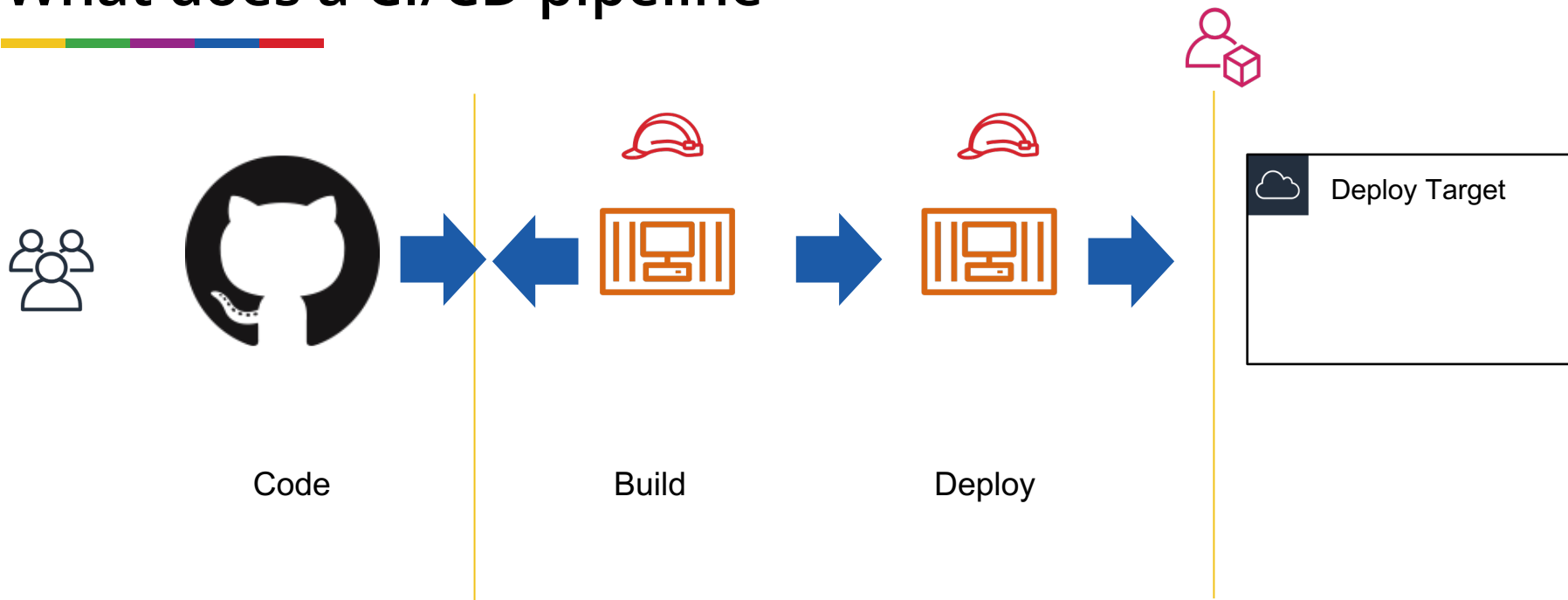


Build



Deploy

What does a CI/CD pipeline



The confused deputy

In a picture



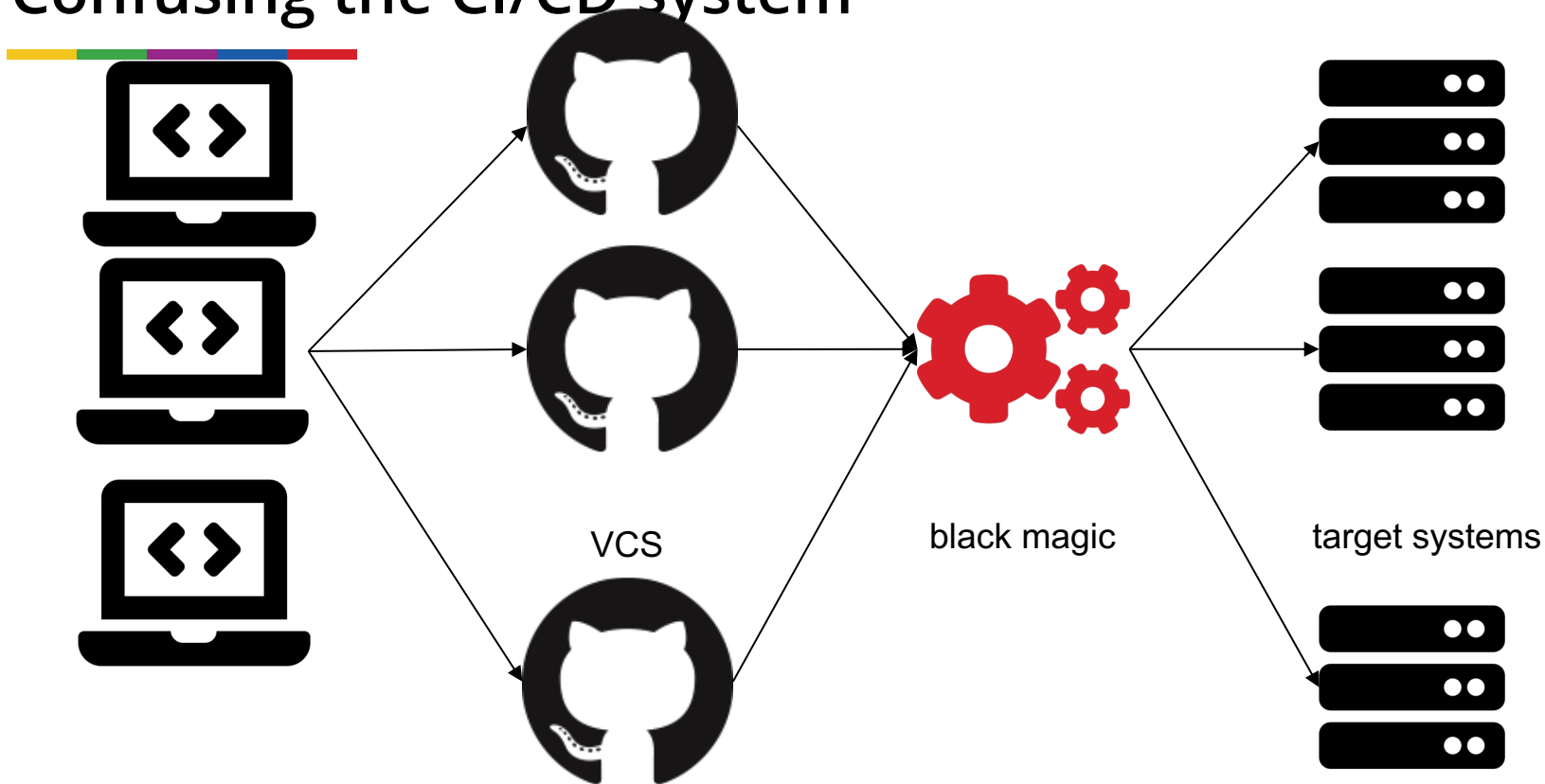
The confused deputy

defined

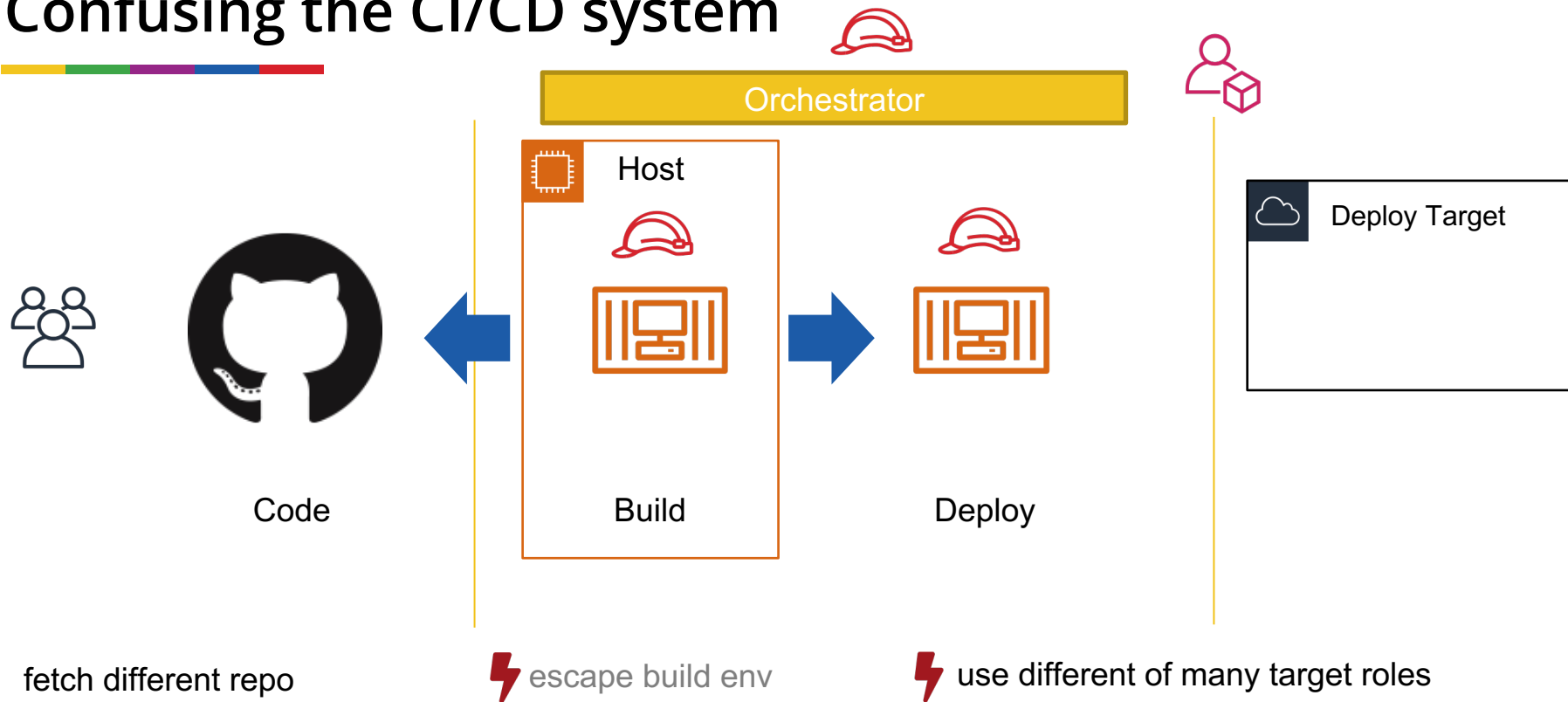
*A confused deputy is a legitimate, more privileged computer program that is **tricked** by another program into **misusing its authority** on the system. It is a specific type of privilege escalation.*

- Wikipedia

Confusing the CI/CD system



Confusing the CI/CD system

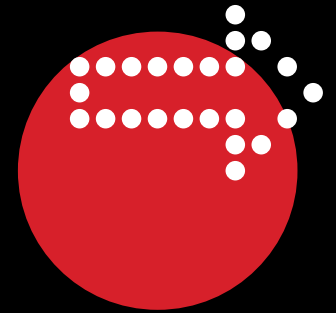


What does a CI/CD pipeline | Problems



Takes code, builds it in a controlled env, deploys it to some environment

- Big, central systems have a huge blast radius
- Acts on behalf of someone with its own identity
- Masks / separates original / triggering user
- Might even make changes to repo
- Effectively allows everyone with push access, access to prod
- All components have own IAM, usually not synced in any way
- Credentials need to be exposed
- Components don't identify each other
- Artifacts are not signed
- build untrusted code (eg. open source)



How to fix?



Step by step

Acting on behalf



- Acts on behalf of someone with its own identity
- Masks original / triggering user

Acting on behalf



Code



Build



Deploy



Acting on behalf



Option 1:

Pass on role with the commit. Afaik not possible right now

Option 2:

- Make all systems identity aware, do not allow to go beyond permissions of pusher
- Remove permission management in between if possible, if needed check out of band
- Reduce confusion possibilities

Acting on behalf | Solution example

- Example solution that is in use at Scout24
- uses common components: GitHub, Jenkins, AWS
- sorry for the complex graphic



AWS STS



Jenkins



GitHub



0) Push



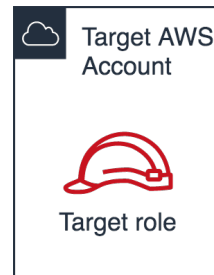
Azure AD



Custom AWS Auth Service
maps Groups to roles in AWS Accounts

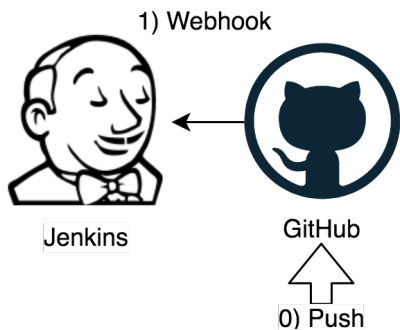


Get Policy of User





AWS STS



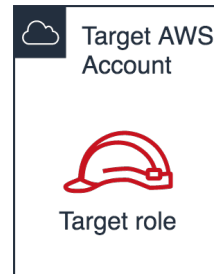
Azure AD



Custom AWS Auth Service
maps Groups to roles in AWS Accounts

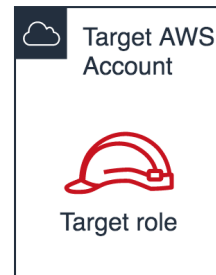


Get Policy of User





AWS STS



2) create ephemeral Agent

1) Webhook



Jenkins



GitHub



0) Push



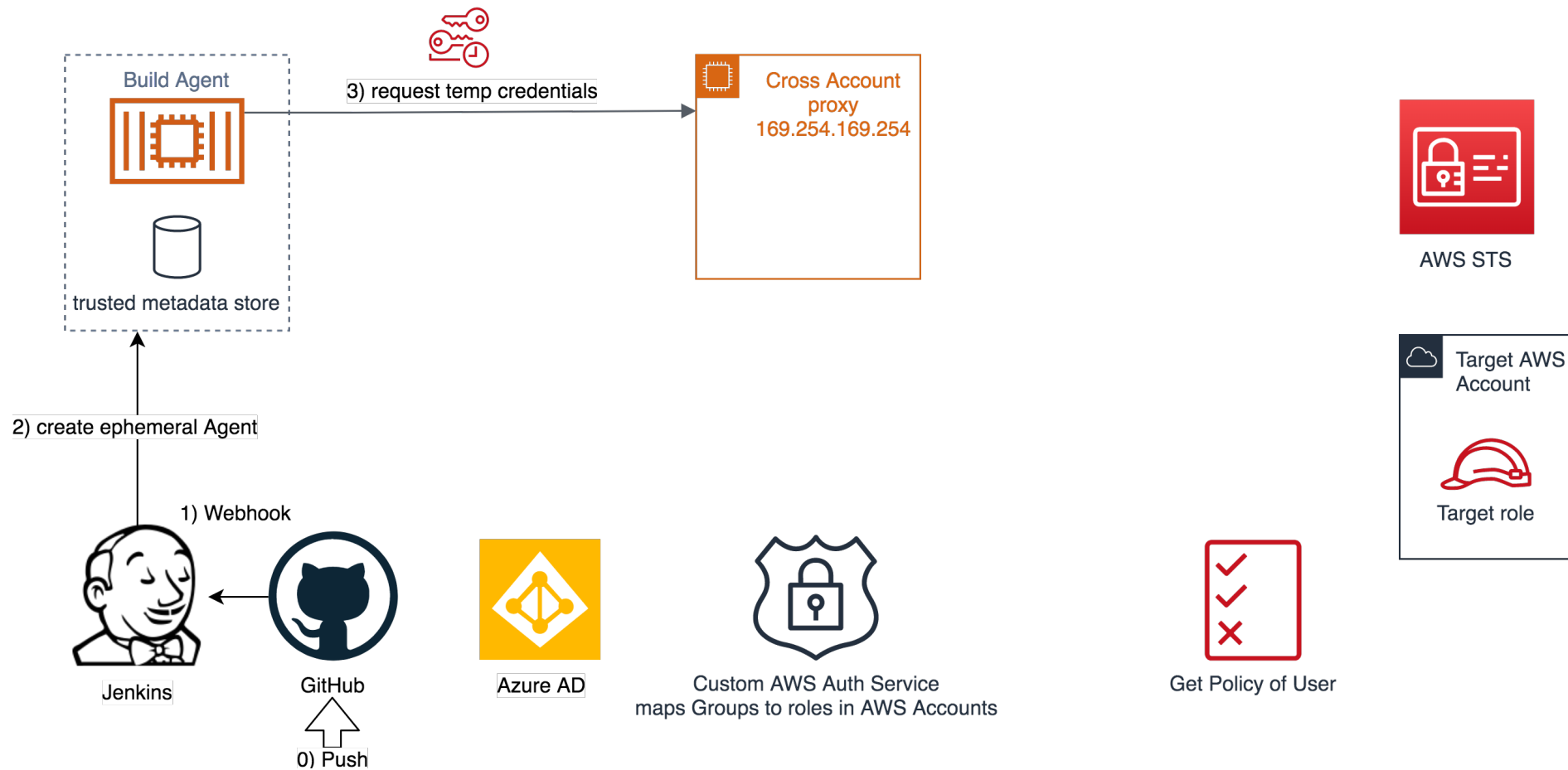
Azure AD

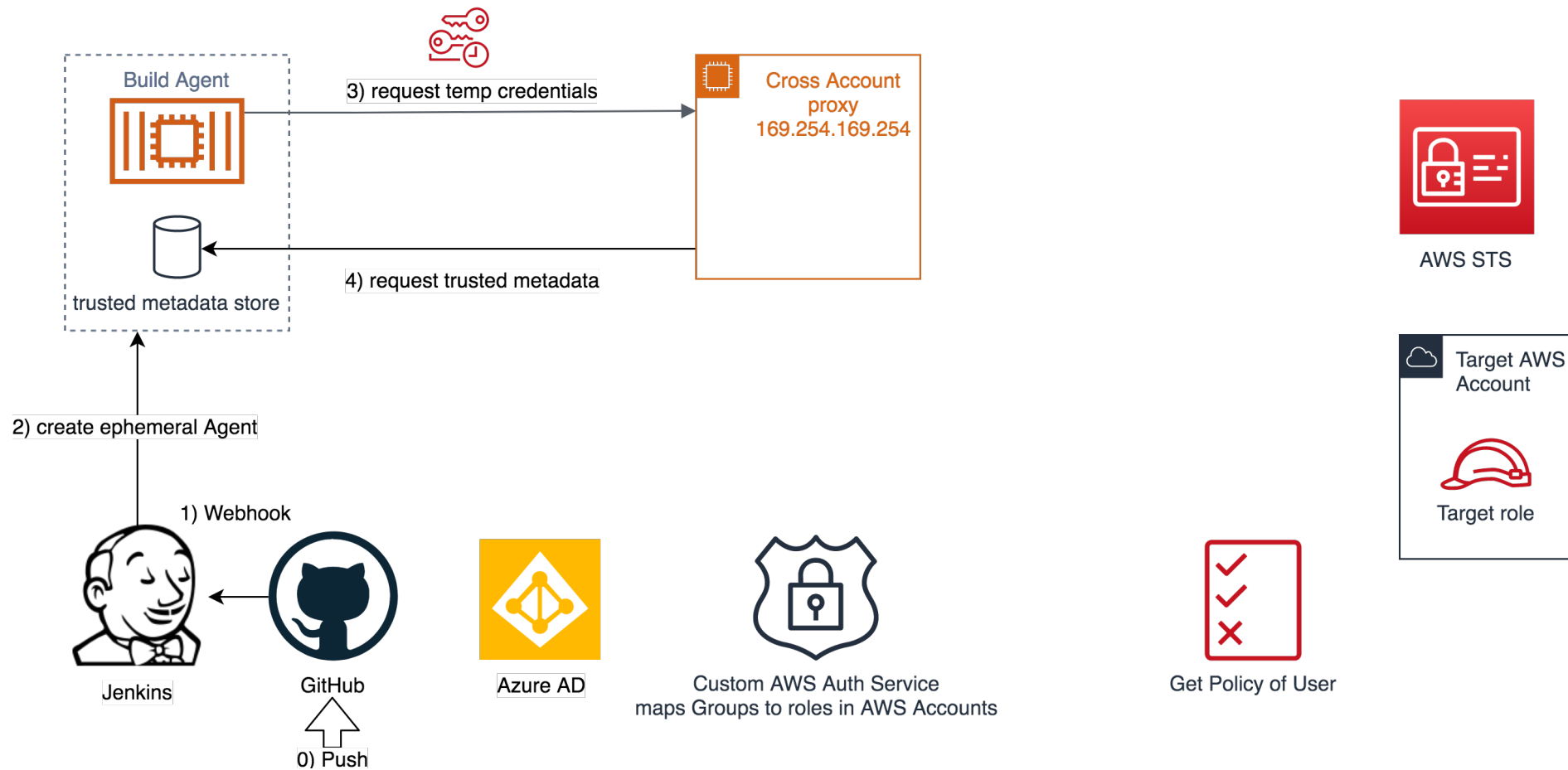


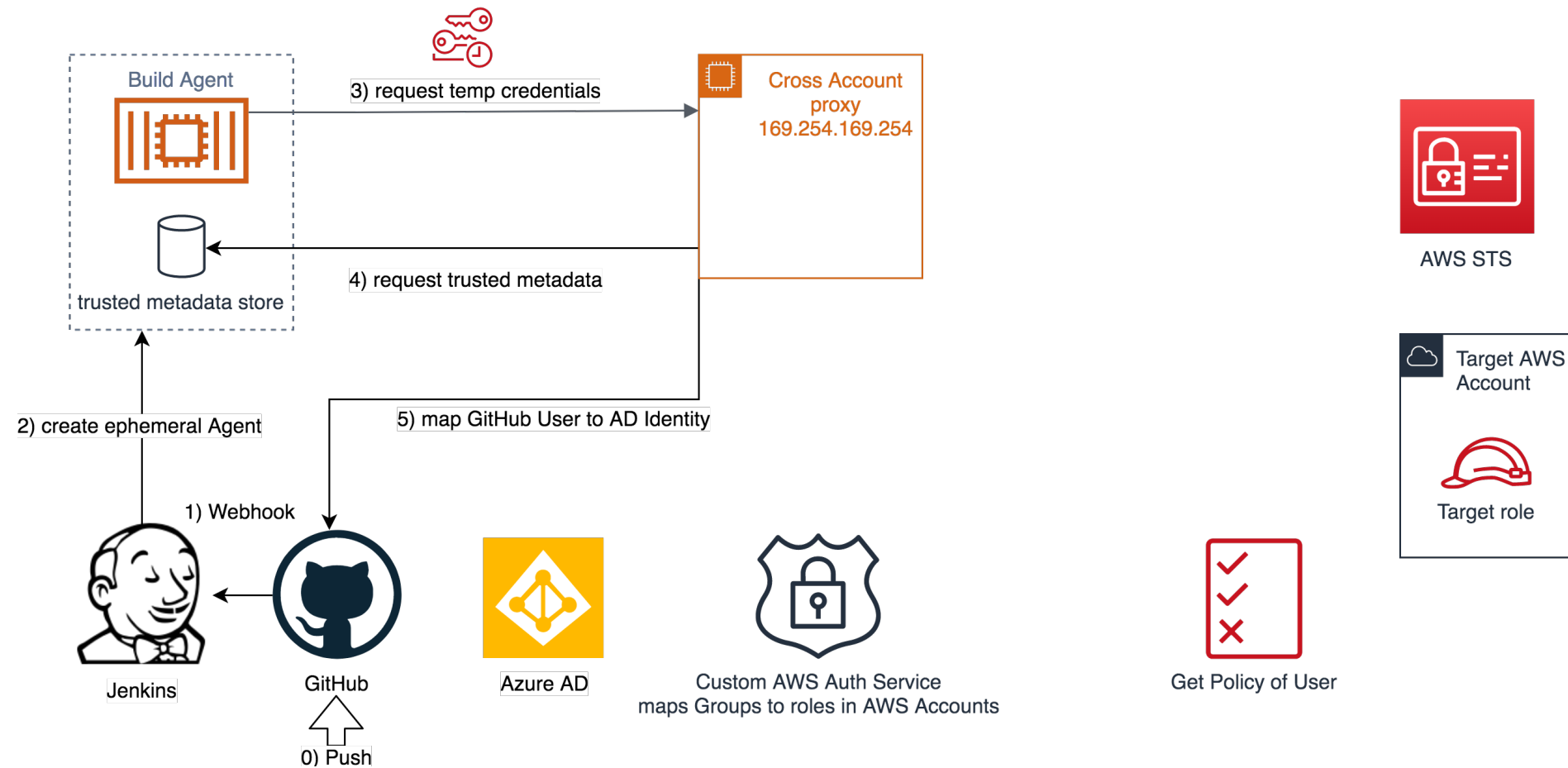
Custom AWS Auth Service
maps Groups to roles in AWS Accounts

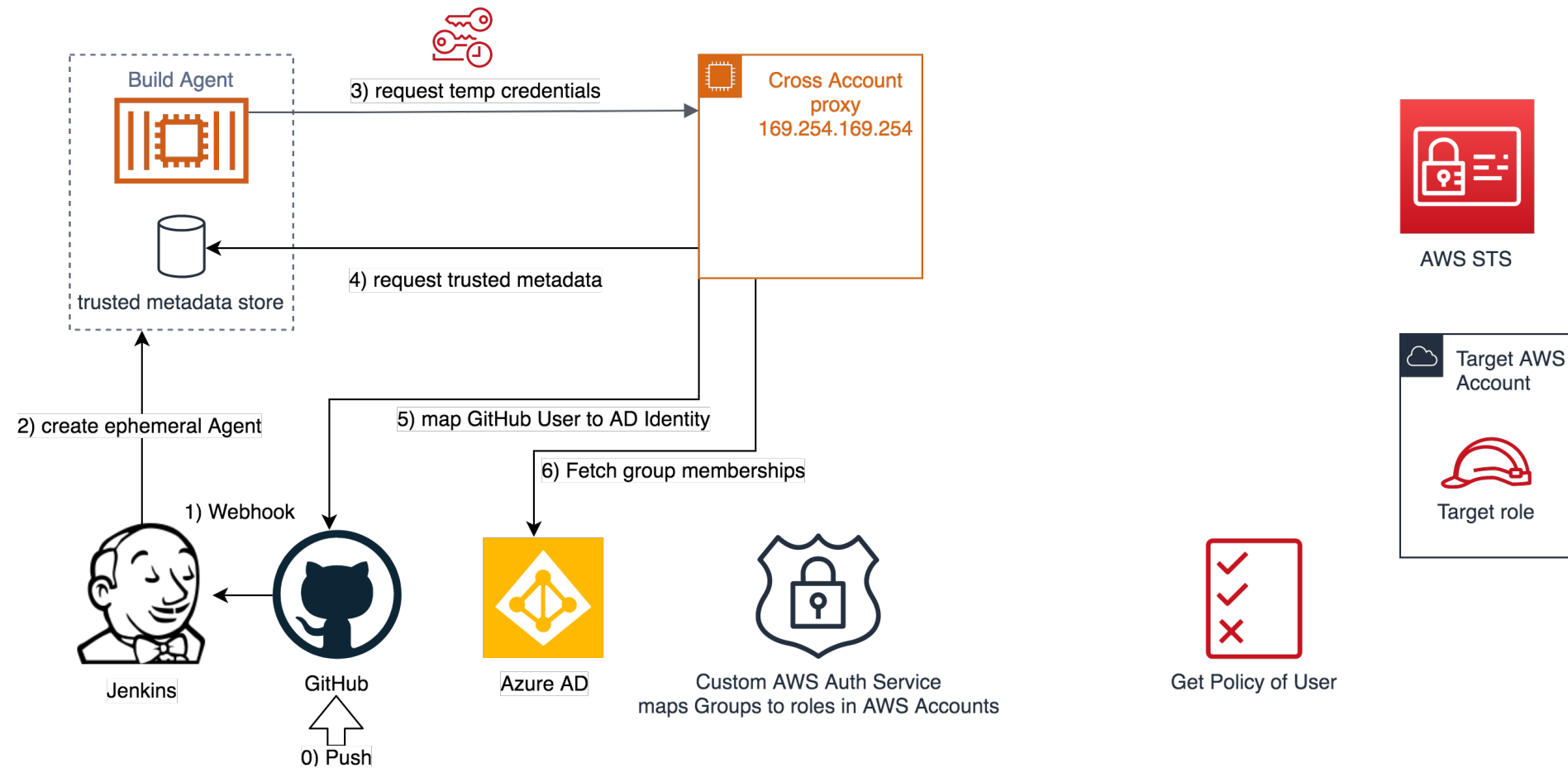


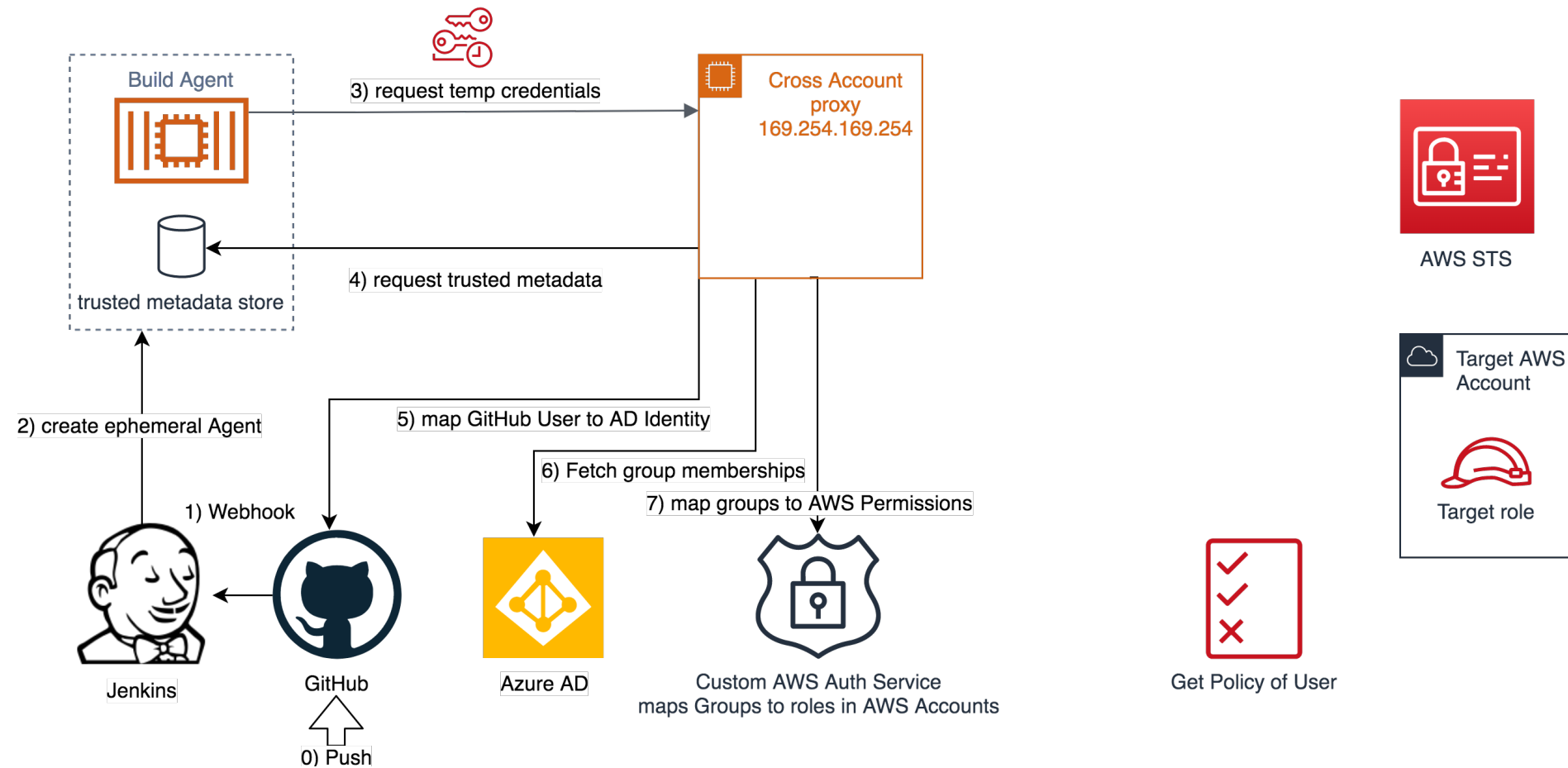
Get Policy of User

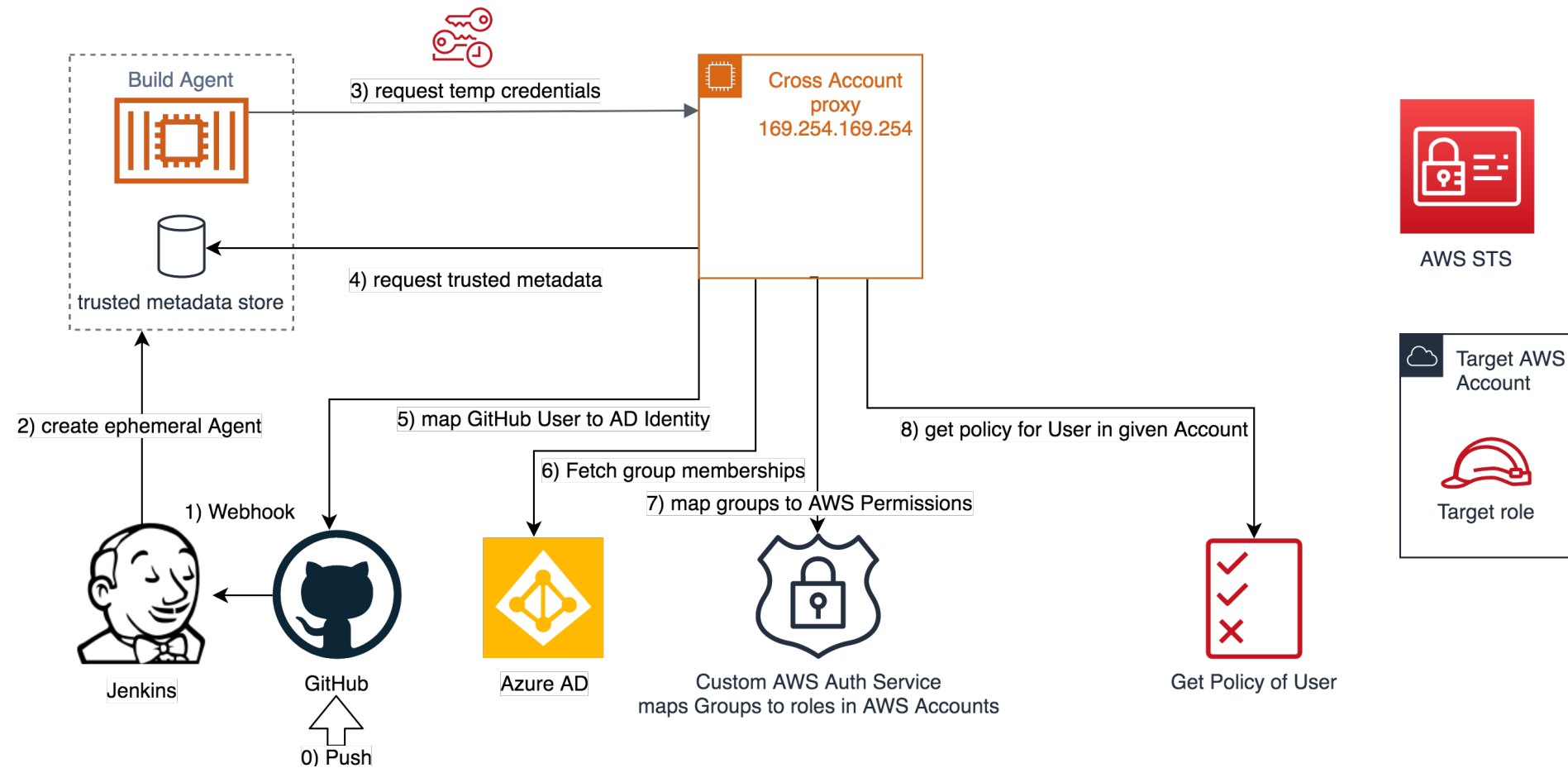


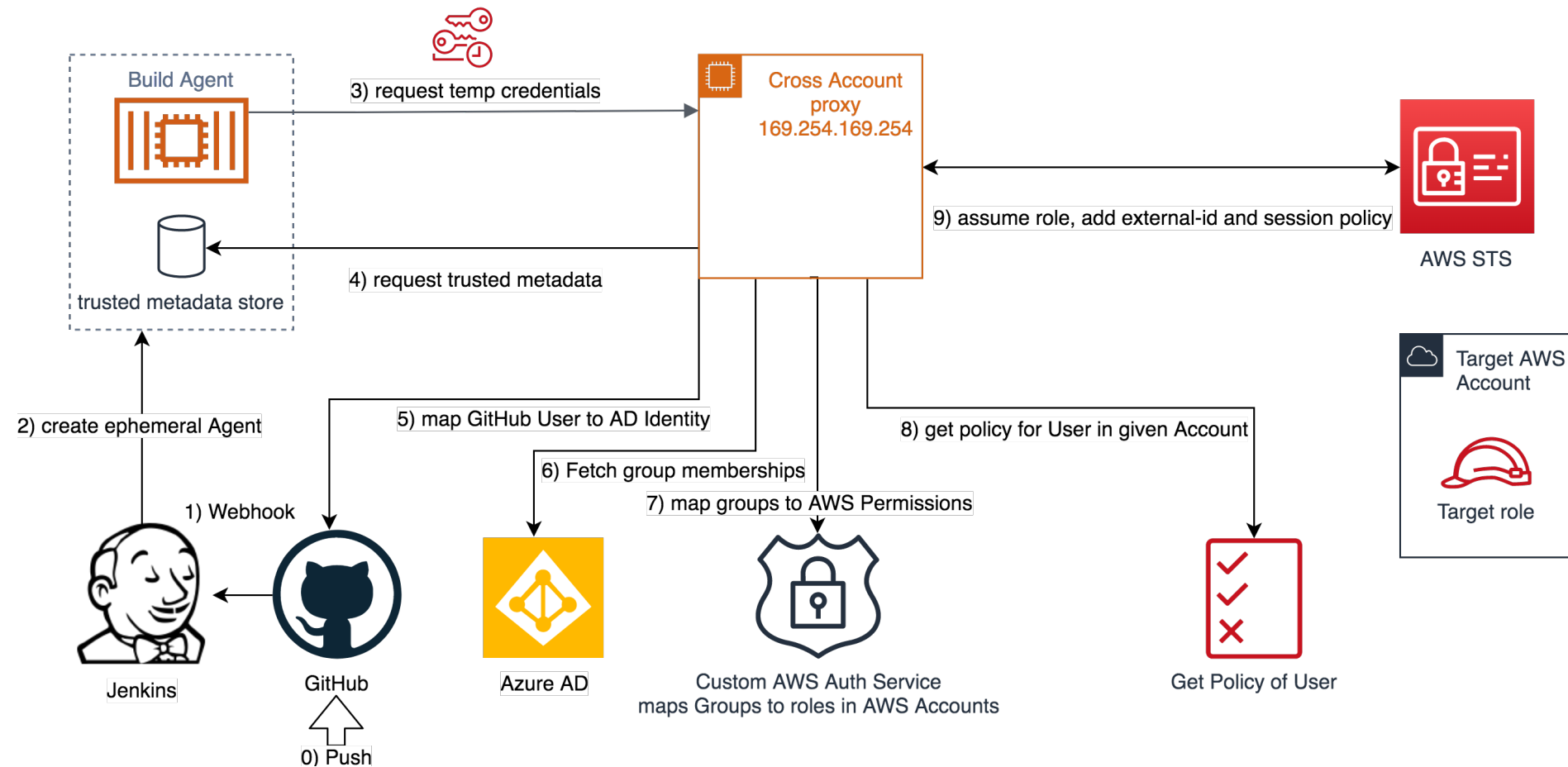


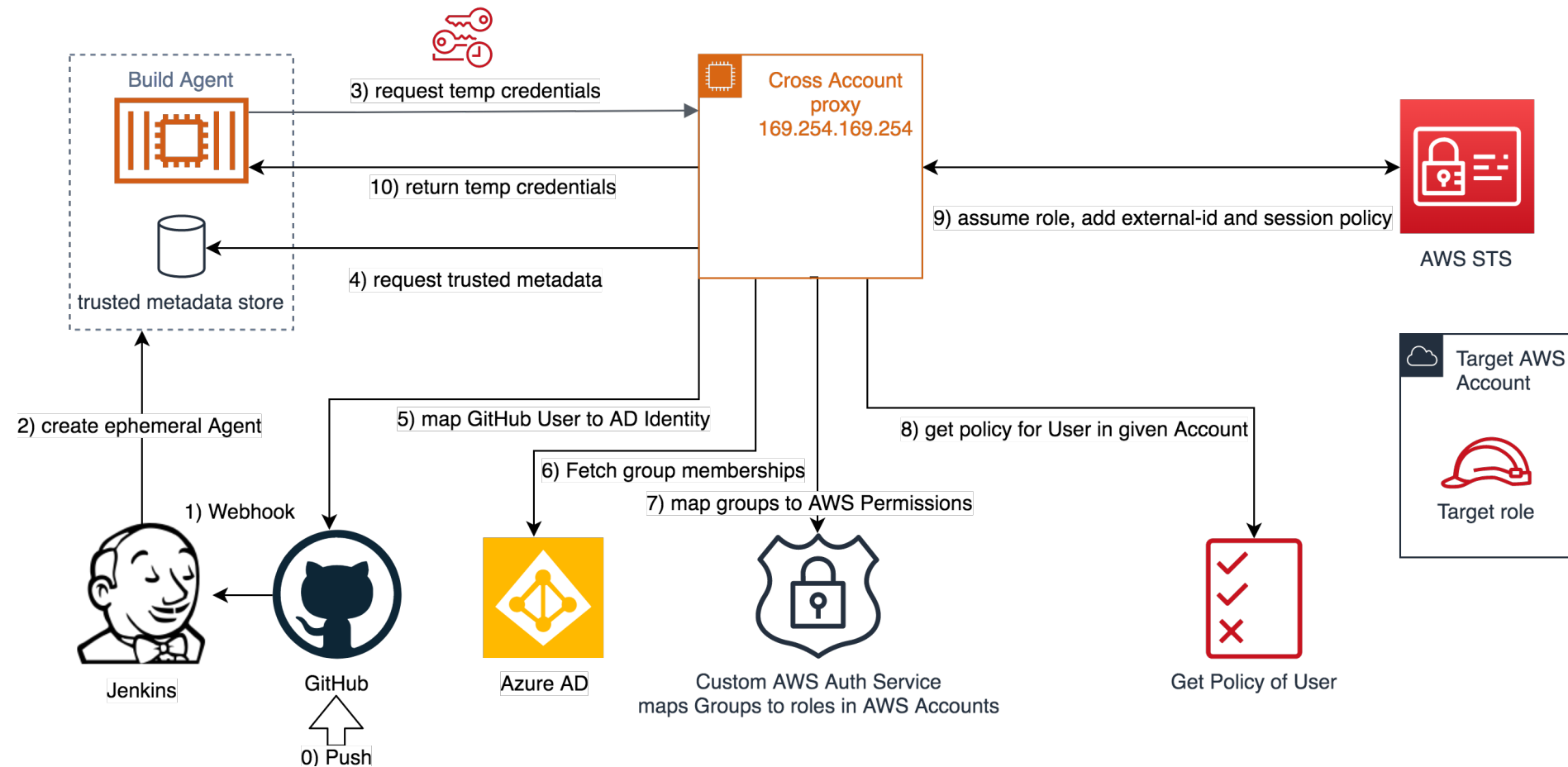












Pro & Con of solution



pro

- one identity used
- out of band check of permissions of user
- user can not gain additional rights
- transparent to existing tools
- target role can restrict to repo
- credential life-time can be very short
- Things can be traced back to user due to Session Name containing the ID

con

- reduced user management, but still options for confusion
- complexity increases
- creator of IAM role needs to ensure external ID check
- needs mapping of git users to roles in AWS accounts
- might not work with other platforms

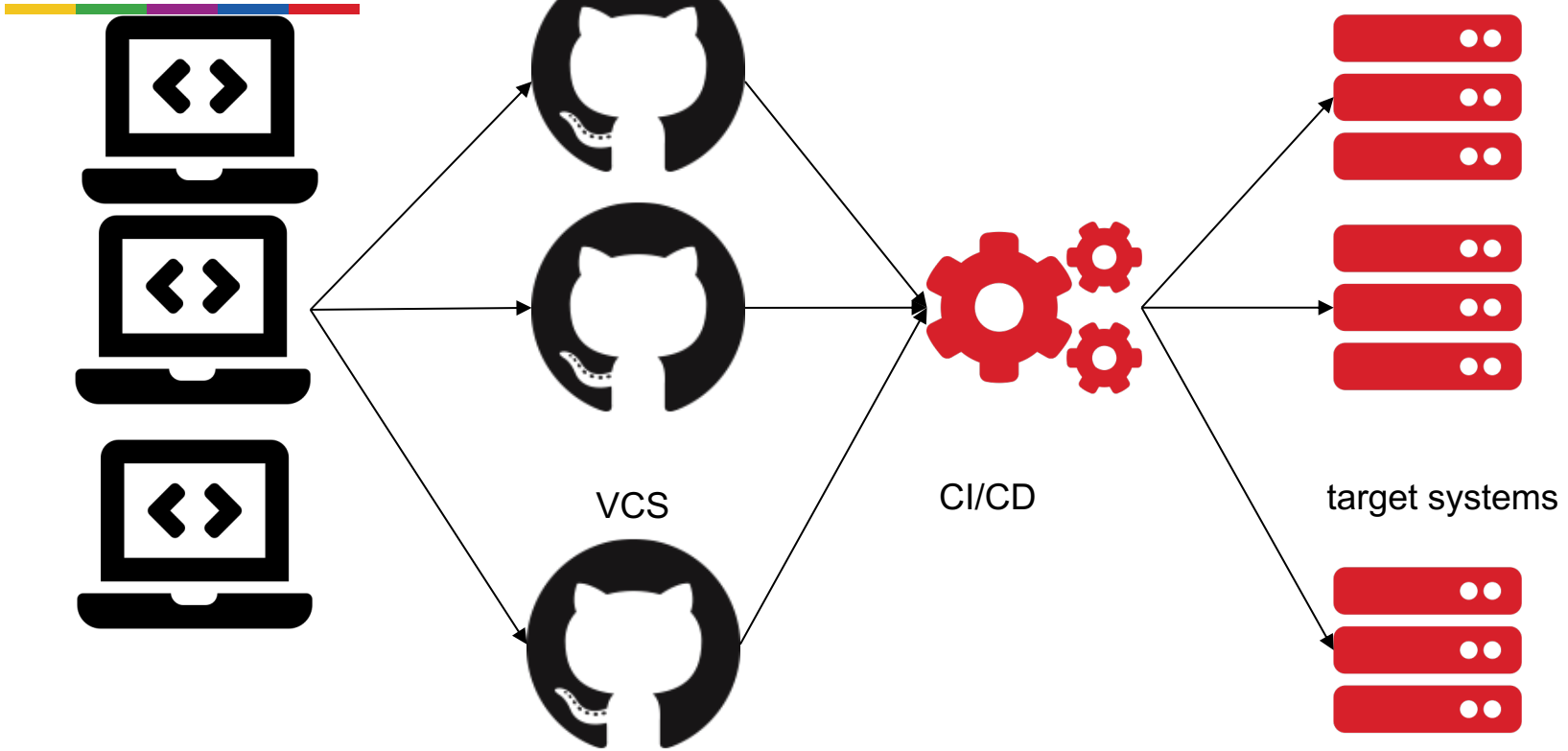
Blast radius



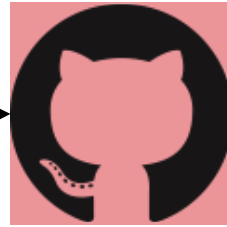
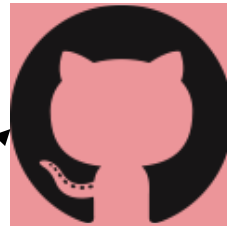
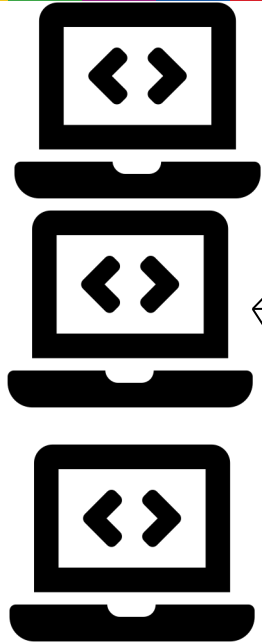
Big, central systems have a huge blast radius

- Gaining access to one component gives access everywhere
- Outages affect everyone

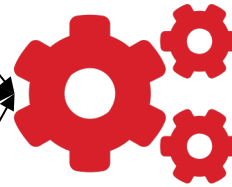
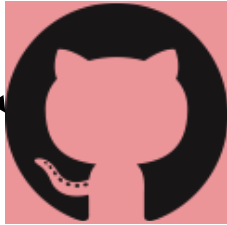
Blast radius



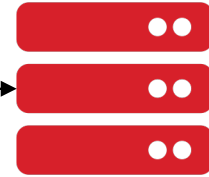
Blast radius



VCS



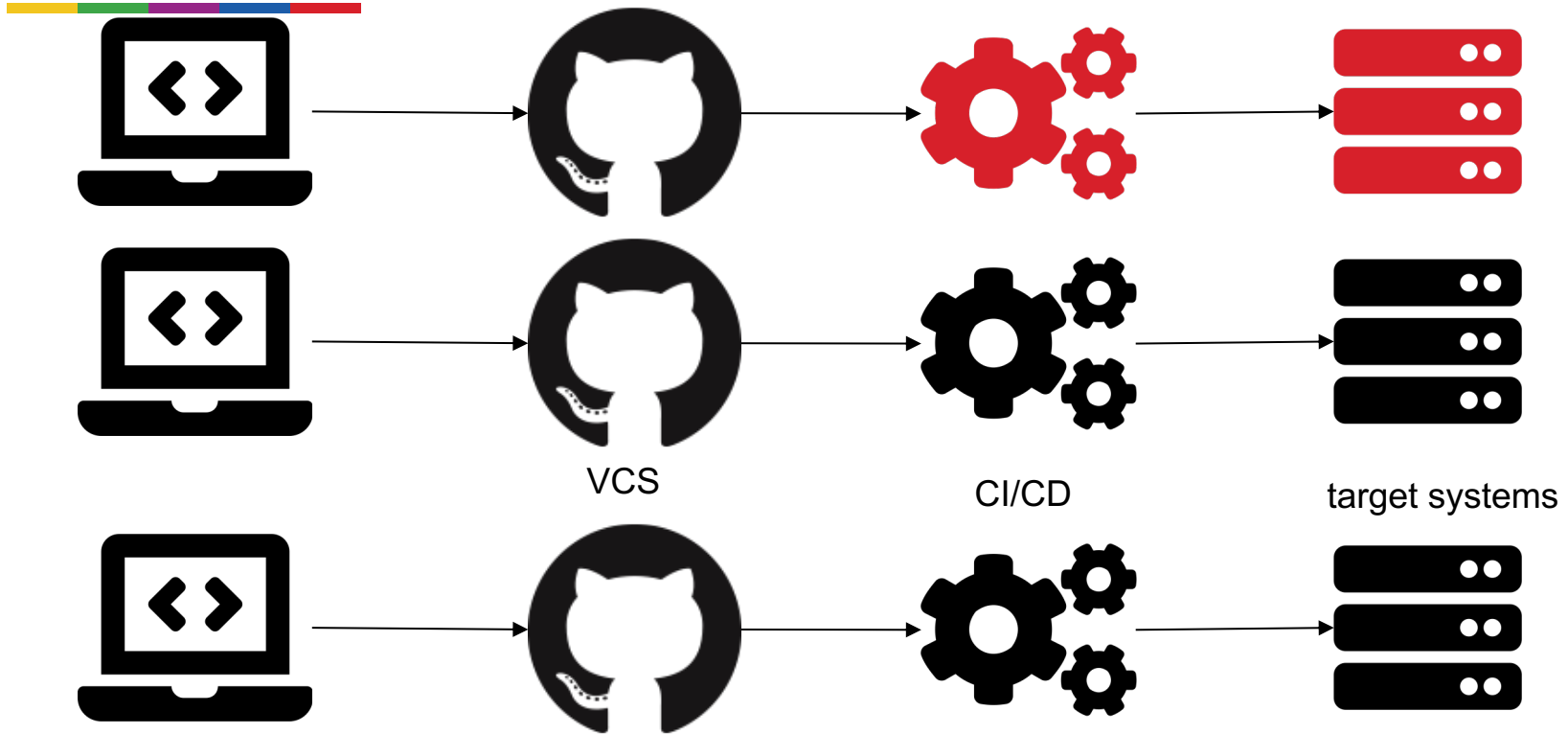
CI/CD



target systems



Blast radius



Blast radius

- Segment into many small independent systems
- Automate / standardize these as much as possible

Scout24:

Had one huge build system for each AutoScout24 and ImmoScout24

Now: over 100 small but automated and standardized instances with limited scope

Exposed credentials



- We need credentials
- Credentials could be echoed
- Credentials are sent to a malicious third party
- Credentials could be stored somewhere and used in other contexts

RESEARCHERS FOUND BACKDOOR IN PYTHON LIBRARY THAT STEAL SSH CREDENTIALS

Share this...



Recently we saw an attempt to hide a back door in a code library, and today there is a new case. This time, **information security** experts found the backdoor in a Python module.

In the SSH Decorator module (ssh-decorate), created by the Israeli developer Uri Goren, which is a library for handling SSH connections from the Python code.

Exposed credentials



- There will always be a need to expose credentials to build & deploy
- Trust to 3rd party dependencies whole topic itself – but locking helps

Reduce impact of stolen tokens:

- limit scope to what's really needed
- rotate very often (at least hourly)

Pushing from within CI/CD

- Whole identity model bases upon actual git users
- No way to track / trace changes done by machine users
- Often non-scoped credentials are in use: CI/CD system can push anywhere 🤖

[maven-release-plugin] prepare for next development iteration



Teamcity committed on Feb 16, 2018

[maven-release-plugin] prepare release parent-0.42



Teamcity committed on Feb 16, 2018

Pushing from within CI/CD



- Don't
- Find alternative strategies (eg release via Tags)

If you have to:

- Get user & repo scoped credentials
- handling follow up actions as initiated by pushing user
- alternative: don't run if you can't identify pushing user

Push to Git == Full Access to prod

- Everyone with push access to your repo can access prod (and more)
- PR builds from forks can be dangerous as source is unclear

Push to Git == Full Access to prod

- Deal with it
- Use same auth source for everything: Sync users and groups
- Be careful with (fork) PR builds. Never give them access to prod credentials / don't build them
- Branch protection & mandatory code reviews / few trusted writers
- Regular reviews of permissions
- Use permissions of pusher for following steps (not committer)

☒ Restrict who can push to matching branches

Specify people, teams or apps allowed to push to matching branches. Required status checks will still prevent these people, teams and apps from merging if the checks fail.

🔍 Search for people, teams or apps

People, teams or apps with push access



Organization administrators, repository administrators, and users with the Maintain role.

These members can push when required status checks pass.



Scout24/delivery-engineering
4 members



☒ Require pull request reviews before merging

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.

Required approving reviews: 2 ▼

☒ Dismiss stale pull request approvals when new commits are pushed

New reviewable commits pushed to a matching branch will dismiss pull request review approvals.


☒ Require review from Code Owners

Require an approved review in pull requests including files with a designated code owner.

☐ Restrict who can dismiss pull request reviews

Specify people or teams allowed to dismiss pull request reviews.

And now?

- 
- Are we there yet? – Sorry, nope
 - Tackle easy things first
 - Build capabilities to link all permissions to identities
 - Get rid of separate permission management wherever possible
 - Improve step by step
 - Talk about it

Read on

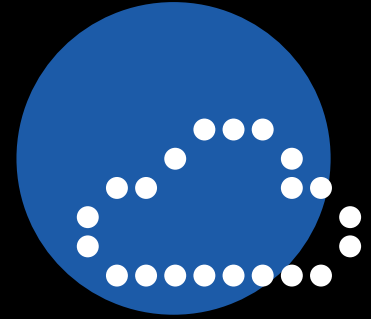


How does Scout24 handle GitHub access?

- <https://www.youtube.com/watch?v=2psQDVIMGlc> | Talk at GitHub Satellite by Jannet Faiz

Detailed Info about Scout24 CI/CD system (mid 2018)

- <https://www.slideshare.net/PhilippGarbe1/run-jenkins-as-managed-product-on-ecs-aws-meetup>



Privilege escalation in build pipelines



kthxbye

Andreas Sieferlinger

Senior Cloud Platform Engineer

@webratz

