Securing Cloud Applications and Their Credentials

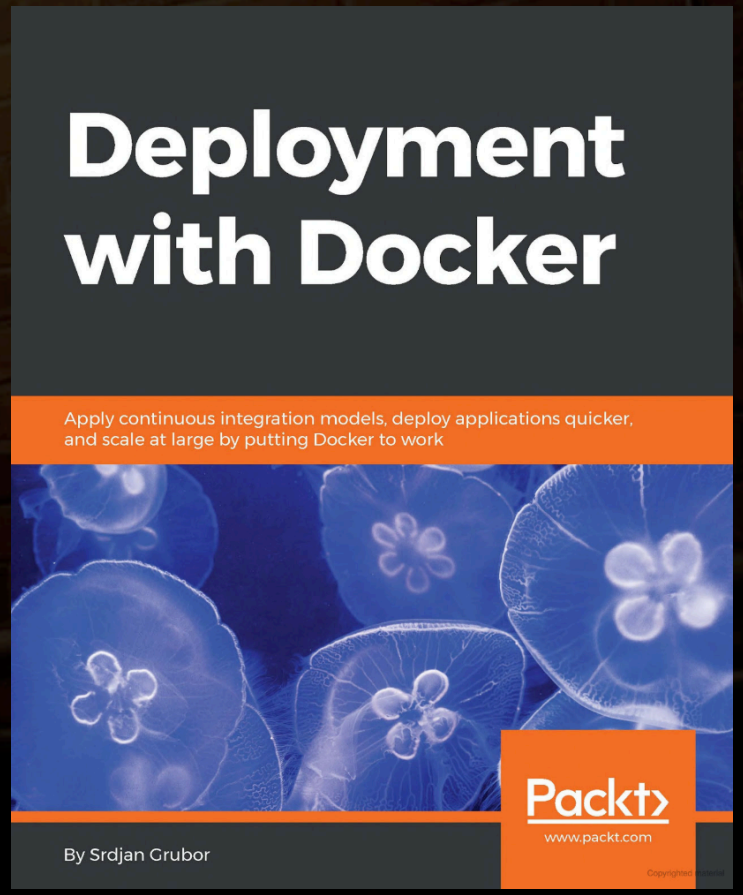
Srdjan Grubor

Strange Loop 2019

About Me

Srdjan Grubor

- R&D Software Engineer @ CyberArk
- Working on next-gen identity platforms
- Author of “Deployment with Docker”
- Docker Certified Associate
- @sgnn7 / sgnn7@sgnn7.org




Why should we care?

Why should we care?

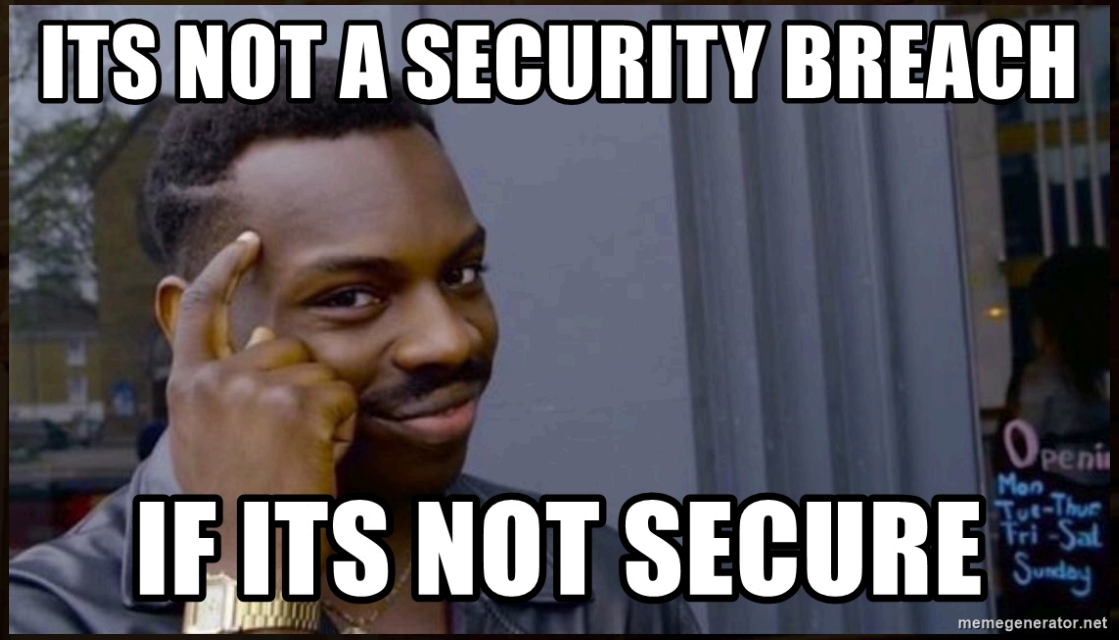
Organizations that embrace velocity without managing risk *will* get breached!

Security

- These is no perfectly secure system.
- Really, there is **no** perfectly secure system…
- The *only* secure computer system is one that is turned off!

Security Wants vs Needs

- Chance of breach?
- Potential impact of breach?
- Friction of use?
- Overhead?



Security - The Hidden Cost

15 SW. Eng. * $51/hr * 10 logins/day * 3min/2FA login * 260 days/year

2000 hours, $100k/year (!!)

Ideal Security Setup

- Warranted*
- Seamless
- Composed of multiple layers
- Automated
- Follows principles of least privilege

Containers

- Add additional layers of security*
- Relatively low friction
- Portable
- Scalable
- Reproducible

Containers: OOTB Security

- Cgroups
- Namespaces
- Single process runner
- Host kernel
- `—cap-drop`

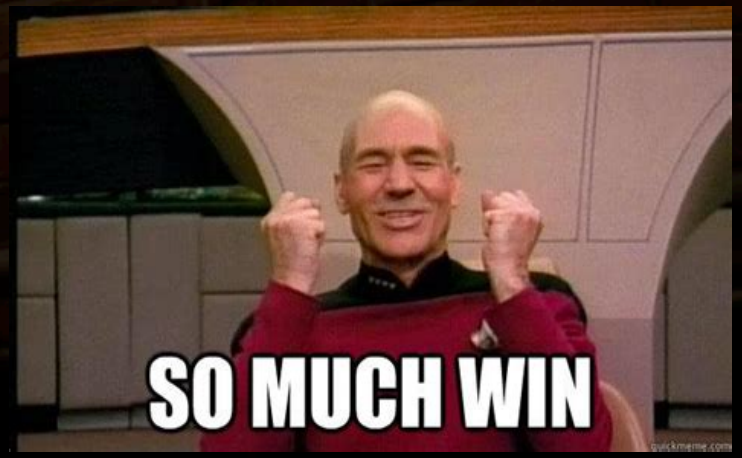
Containers - Additional Perks

- Discretionary Access Control (DAC)
  - Filesystem permissions
- Mandatory Access Controls (MACs)/LSM
  - AppArmor
  - Seccomp
  - SELinux
- Chroots
- Copy-on-write filesystem
- Image hash verification


Containers are very secure
…but it’s too easy to sabotage yourself
😢

Avoiding Container Pitfalls

- Never mount `/var/run/docker.sock` into the container*
- Keep up with your base images updates
- Run container services as a limited user, where possible*
- Keep an eye on:
  - Overly-broad volume mappings
  - Intra-pod communication attack vectors
  - Repository TLS exceptions*
- Use static image analysis tools (Docker Bench, Clair, etc.)
- Use private image repositories when feasible
- Use some combination of Seccomp / AppArmor / DAP



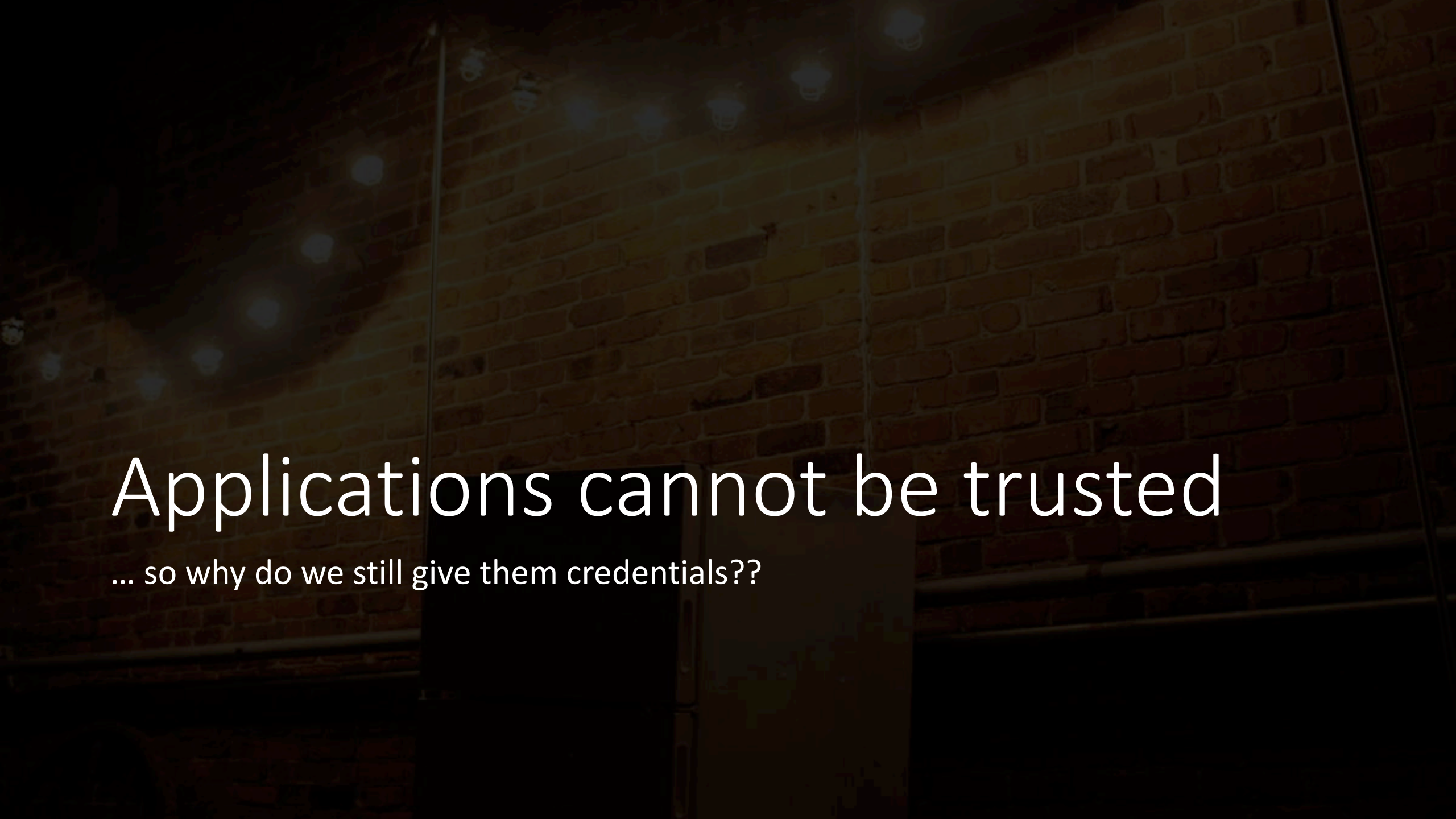
Modern Apps

Less:

- Reinventing the wheeI
- In-house code
- Understanding of underlying logic
- Control over abstractions

More:

- Configuration
- External libraries
- Unverified code
- **Exfil potential**
- **Malicious code**



Applications cannot be trusted

… so why do we still give them credentials??

State of Things - 2019

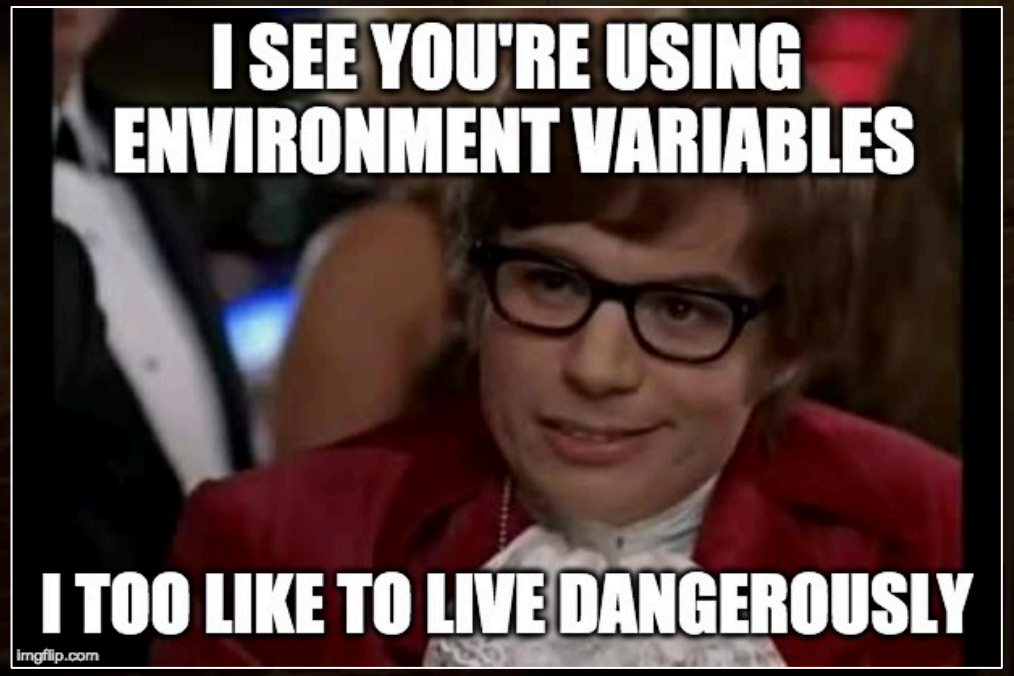
Developer

Ops/Security

Credentials And Secrets

“74% Of Data Breaches Start With Privileged Credential Abuse” – Forbes, Feb 2019

Securing Containerized Credentials



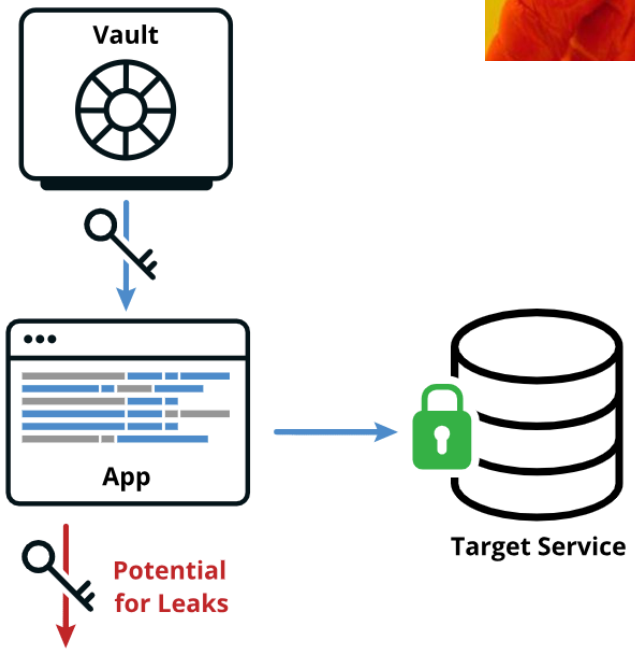
- **Popular methods for providing secrets to containers**
  - ~~Via hardcoded credentials~~
  - Via environment variables
  - Via volume mount
  - Secrets encryption
- **Main challenges**
  - Secrets can be easily exposed
  - No runtime authentication process of the calling container
  - Lack of segregation of duties
  - No audit trail

Credential Management

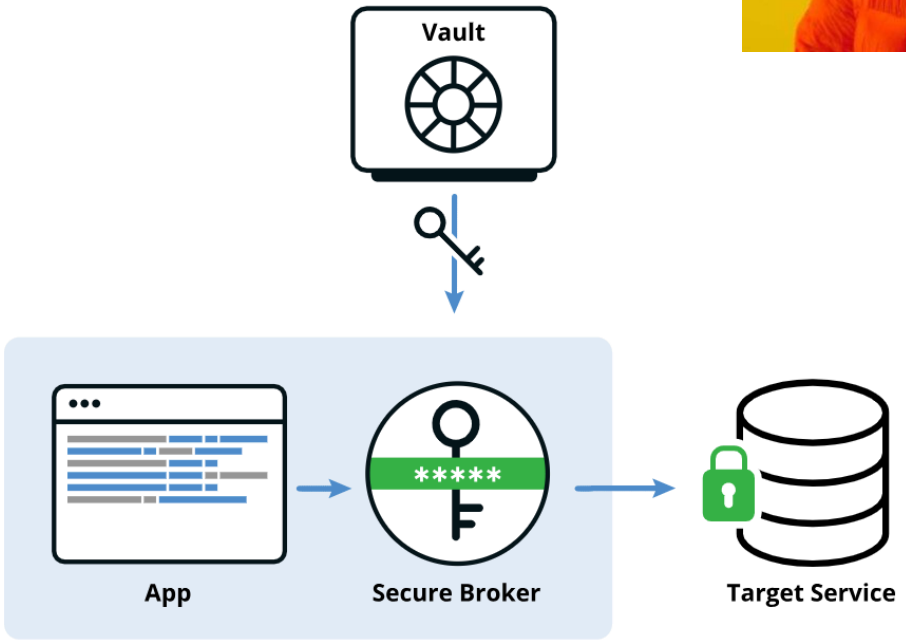
- **Vaults**
  - Centralized storage location for your credentials
  - RBAC/ABAC enforcement
  - Rotation of credentials
- **“Authenticators”**
  - Platform-native identity validation (SPIFFE / K8s API / etc)
- **Last-mile “credential enclaves”**
  - Encapsulates/isolates app from the credentials
  - Prevents the app from ever knowing any sensitive data
  - Separates security concerns from development concerns

Secretless (secretless.io)

The Old Way



With Secretless Broker



Secretless

Proxied connections w/o direct app access to secrets

- Frees devs and apps from responsibility of managing secrets
- Reduces the threat surface of secrets
- Handles rotation transparently
- Does not change how clients connect to services
- Allows use of standard libraries and tools
- Open source at github.com/cyberark/secretless-broker!

Questions?

Srdjan Grubor

sgnn7@sgnn7.org

@sgnn7

More information

- Secretless : https://secretless.io
- Conjur OSS: https://conjur.org
- Docker Bench: https://github.com/docker/docker-bench-security
- Clair: https://github.com/coreos/clair
- Seccomp: https://docs.docker.com/engine/security/seccomp/
- AppArmor: https://docs.docker.com/engine/security/apparmor/
- Perfect security implementations:
  - https://theonion.com
  - https://youtube.com/watch?v=dQw4w9WgXcQ