Build your own multi-user password manager using open source software

Michael Vieau, CISSP, CEH
Kevin Bong, GSE, PMP, QSA, GCIH, GCIA, GPPA, GAWN, GCFA, GSEC, CISSP

DerbyCon 2019

SIKICH®

ABOUT US

Sikich is a national technology consultancy with a single-minded focus on improving business performance by deploying best-fit technology solutions. We help our clients understand “what could be,” help them set priorities, and take responsibility to deliver transformative digital strategies.

SIKICH®

CYBERSECURITY

OUR SERVICES

We help customers understand security and compliance risks, how to avoid those risks, and what to do should a security incident occur.

FORENSIC **INVESTIGATIONS**

VULNERABILITY **SCANNING**

RISK **ASSESSMENTS**

PENETRATION **TESTING**

IT **AUDITS**

REMEDIATION

SECURITY **AWARENESS TRAINING**

SIKICH®

SIKICH.COM

THE SITUATION

- Automated build process for pentest drop boxes deployed to client sites
  - AutomaEach drop box has a strong, unique password
  - ted tools need to set and store the password
  - Pentesters need to access the password
- Keepass currently being used
  - Difficult to automate entry of new passwords
  - Does not support multiple users concurrently
- We’re smart, lets build something, it will be easy!

SIKICH®

THIS WILL BE SO EASY

- Let's:
  - Make it extra cool and present it at DerbyCon!
  - Use public-key cryptography to protect everything
  - Make it open source
  - Build it using scripts/wrappers around GPG so it's easy to vet
- This will be so easy, why hasn't anyone done it before?

SIKICH®

THE DESIGN

- Create a key pair for the password server
- Encrypt each password with the server's public key
  - This allows an untrusted source to submit a new password without looking up other passwords
- Protect the server's private key by encrypting it using each application user's public key
- All application users password protect their own private keys to enforce multi-factor authentication (MFA)

SIKICH®

THE BATTLE

- Design disagreements arose very quickly
  - Database vs. file storage
  - Decrypt on client vs. Decrypt on server
  - No encryption key, one encryption key, one encryption key per stored password
  - Never rotate key, rotate key as needed, rotate key on each use
- All of these decisions were interdependent

SIKICH®

WHERE TO STORE?

- Database
  - Multi-user
  - Fast
  - Helps enforce integrity
- Files
  - No need for a dedicated server

SIKICH®

WHERE TO DECRYPT?

- On client
  - Minimizes security requirements on the server
- On server
  - More control over the system where decryption occurs
  - Fewer secrets in memory on client systems
  - Logging
  - Ability rotate keys in batch

SIKICH®

HOW MANY ENCRYPTION KEYS?

- None – just use the PGP keys of the application users
  - Simple design to explain and justify
  - Drawbacks with complexity of implementation and storage needs
- One – same server public key encrypts all passwords
  - Easiest to add passwords or rotate keys
  - Drawback with a disclosed private key putting all passwords at risk
- Many – unique server public key per protected password
  - More secure – a stolen server private key only exposes one password
  - Drawbacks with complexity of key management and storage needs

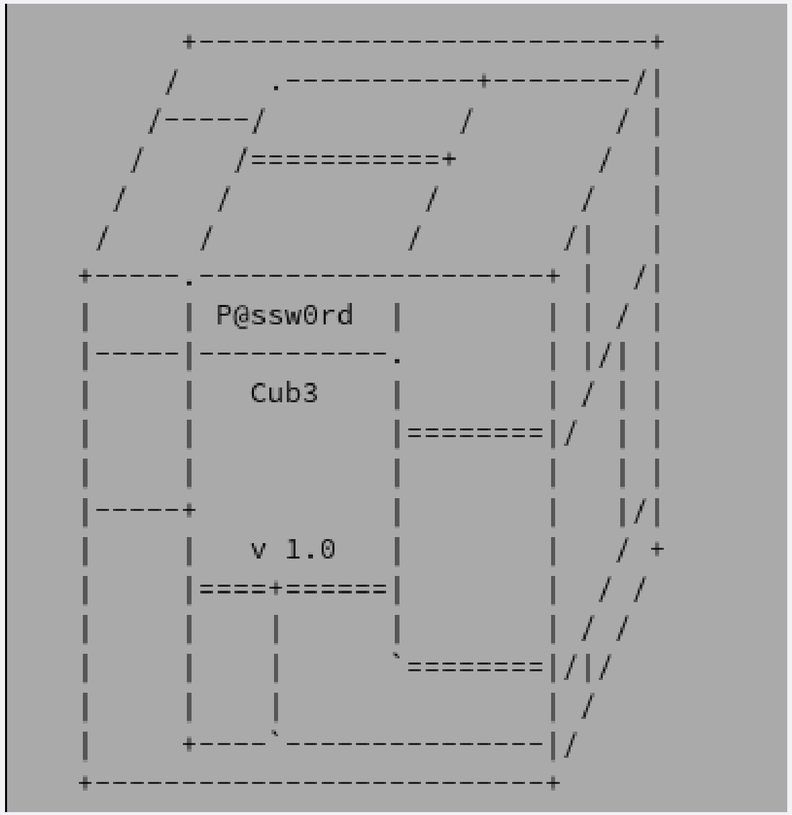
WHEN TO ROTATE ENCRYPTION KEYS?

- Never – rotation would take manual effort if we ever do it
- As needed – build a process we run when an employee leaves
- At each password lookup – set a new key each time the old one is used

THIS IS SO HARD!

- Lots of great takeaways that we can share at DerbyCon
- But let’s just build what we need for our team

THE SOLUTION



- Database storage using MySQL
- All encryption/decryption occurs server side
- Using GPG for encryption/decryption
- Written in Bash script
- Access to the system via SSH
- One server public key used to encrypt all passwords
- Rotate the server public key as needed
- Open source!

https://github.com/TheMayhamLab/PasswordCub3

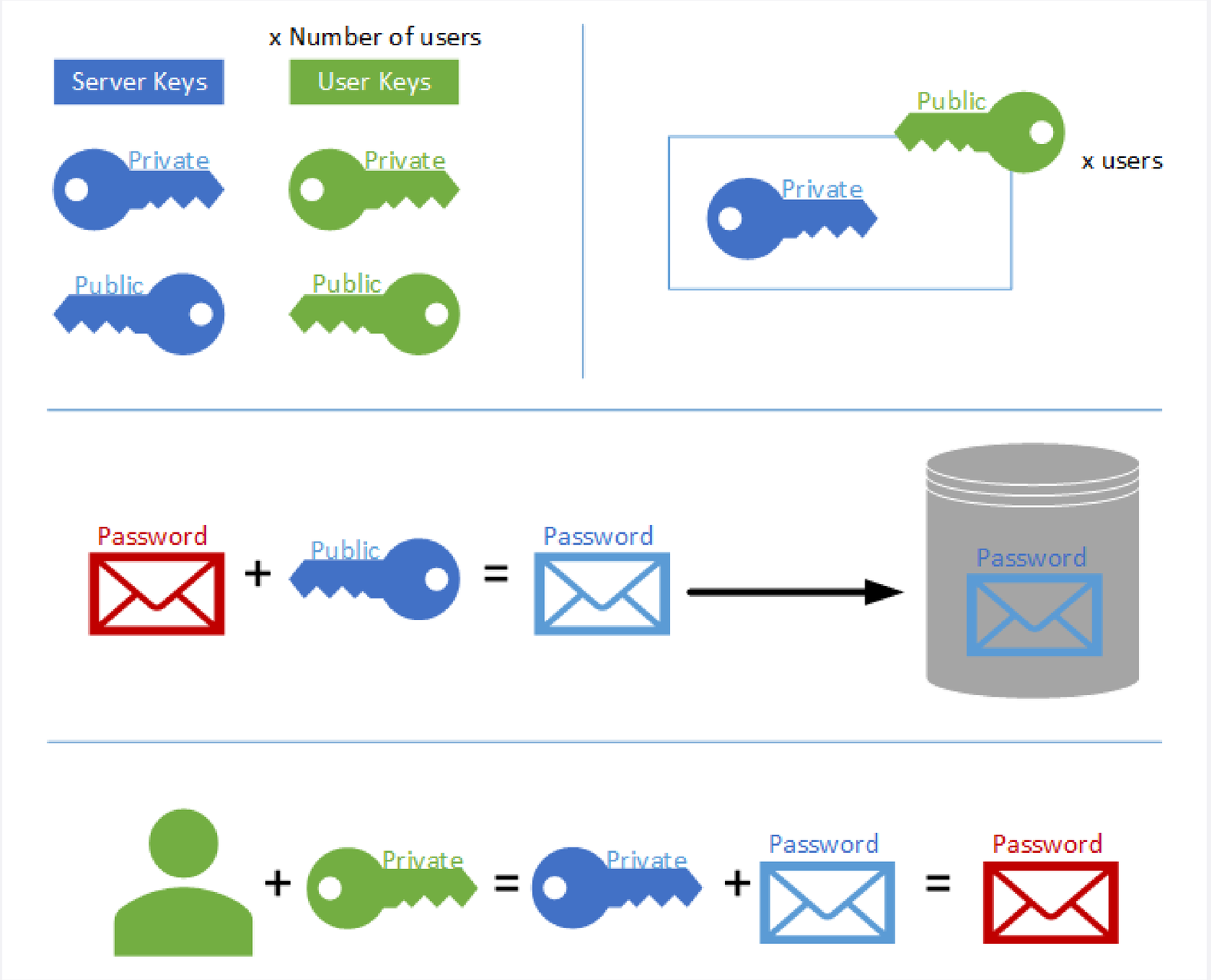
SIKICH®

THE
SOLUTION – DESIGN CHOICES

- Why tempfs vs ramfs
- Why Bash
- Why GPG
- Why SSH

https://github.com/TheMayhamLab/PasswordCub3

THE SOLUTION



SIKICH®

THANK YOU FOR YOUR TIME.

Are there any questions?

Michael Vieau
michael.vieau@sikich.com

Kevin Bong
kevin.bong@sikich.com

SIKICH

SIKICH.COM