

Jumpstart Your Privileged Account Management (PAM) Program

About me

Ken Robertson - @krobert7

- Started IT as a Unix admin in the early 1990's
- Moved into IT security and IAM in 2007
- Worked with my first PAM system in 2010



Jumpstart your privileged account management (PAM) program



Jumpstart your privileged account management (PAM) program



Agenda

- What is PAM?
- Steps to implement PAM
- Maturity over time
- PAM automation
- Account discovery
- Overcome obstacles
- Takeaways



What is a PAM tool?

A PAM tool will have many features used to mitigate risks and control access to privileged accounts.

- Manage credentials
 - Track and monitor privileged activity
 - Control access
 - MFA for authentication
- and many more...



What is a PAM program (service)?

Technology + processes

Building blocks of PAM service

- Governance
- Privileged account inventory
- Privileged user management
- Policies, monitoring and auditing



PAM service design considerations

Adoption must be driven by enterprise policies

- Define what is a privileged account
- Define controls for managing privileged accounts

Fault tolerance and operational excellence

**RESTRICTED
AREA**

**Privileged
Personnel Only**

Preparation for your PAM service

- Promote and advertise!!
- Clean up your Configuration Management Database (CMDB)
- Study your environment, develop the use cases



Preparation for your PAM service

- Define clear goals and create a realistic roadmap
- Don't underestimate the workload of the stakeholders
- Inventory



Design your PAM service

- Life-cycles for users, devices, accounts, and applications
- Make sure you are reducing risks and not checking a box
- Project management



Implementation of your PAM service

- Risk based approach
- There will be quick wins, but full integration ~~could~~ will take years
- Managing application credentials is difficult, especially for the application team



Maintenance of your PAM service

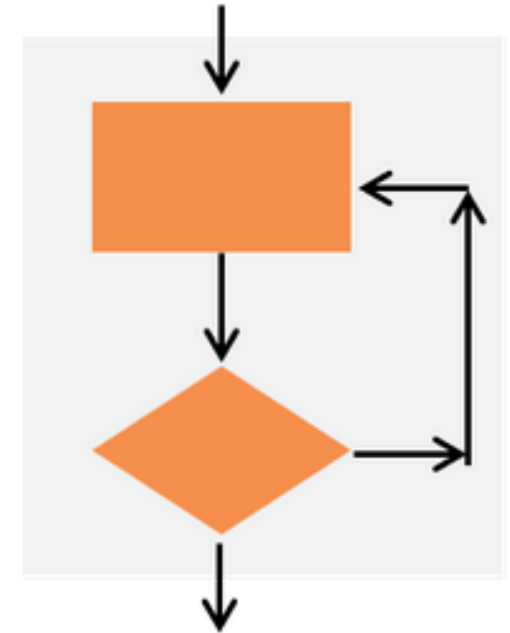
- PAM metrics, monitoring, and audit
- Roadmap - continually improve - quarterly reviews
- Operational maintenance



Maturity over time

- Maturity models
- Build your service and continually improve

REPEAT .. UNTIL ..



Using your maturity model

- Make it your own - customized outcome
- Assess yourself - what is important to my org?



Build a maturity model

https://gitlab.com/kenrobertson/PAM_maturity_model

Example Privileged Access Management (PAM) maturity model - created for Identiverse conference, Jun 2018.

	Level 0 Initial	Level 1 Managed Repeatable	Level 2 Defined	Level 3 Automated	Level 4 Optimized
Characteristics	No Controls, Unpredictable and reactive	Tribe knowledge, often reactive	Proactive, Standards documented	Automated, Consistent, repeatable	Stable and flexible, full automation
Account management	Manual provisioning and deprovisioning	Some accounts automatically provisioned and removed	APIs available for all systems to use in automation	All accounts automatically provisioned and removed	Accounts automatically provisioned with ABAC
Session management and isolation	None	Some sessions managed	Session hosts and jumpboxes available for all environments	All sessions managed with no passwords exposed to users	Required for all segmented networks
Application credential management	None	Some applications with manual credential management	Most applications are covered	Programatic application account management	All applications are in scope and compliant
Session Recording	None	Some privileged sessions recorded and stored	Automated searching of recordings	All privileged sessions recorded	Alerting and integration with your threat detection tools

PAM automation

Build a utility / automation server

- Knowledge in one place
- Automation ideas
 - On and off boarding
 - User management
 - Account discovery
 - Metrics
 - Look at your incident tickets



Account management

- Reduce number of privileged accounts
- Move from personal privileged accounts to a shared model
 - Reduce number of accounts
 - Enforce least privilege principle
 - Remove standing privileges



Account management continued

Principle of least privilege

- create accounts with specific function in mind

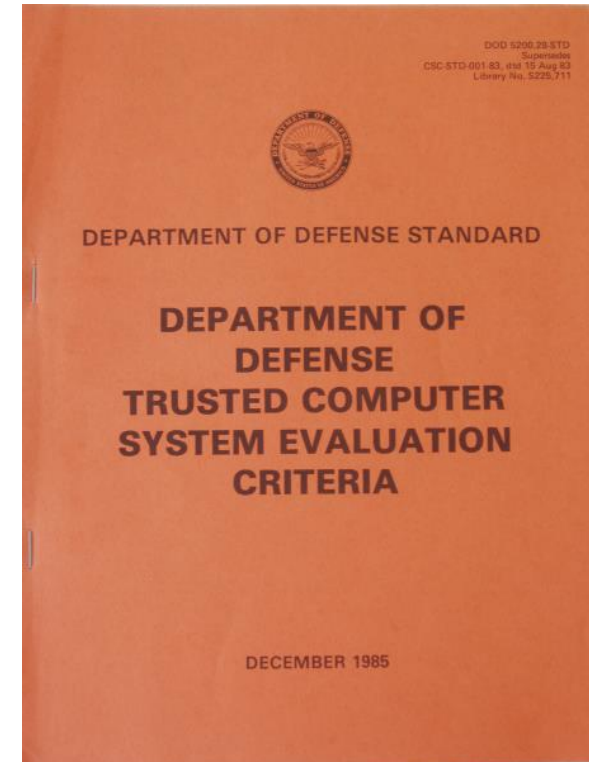


Documentation should be your best friend

Divio system

<https://documentation.divio.com/>

- Tutorials
- How-to guides
- Explanation
- Reference



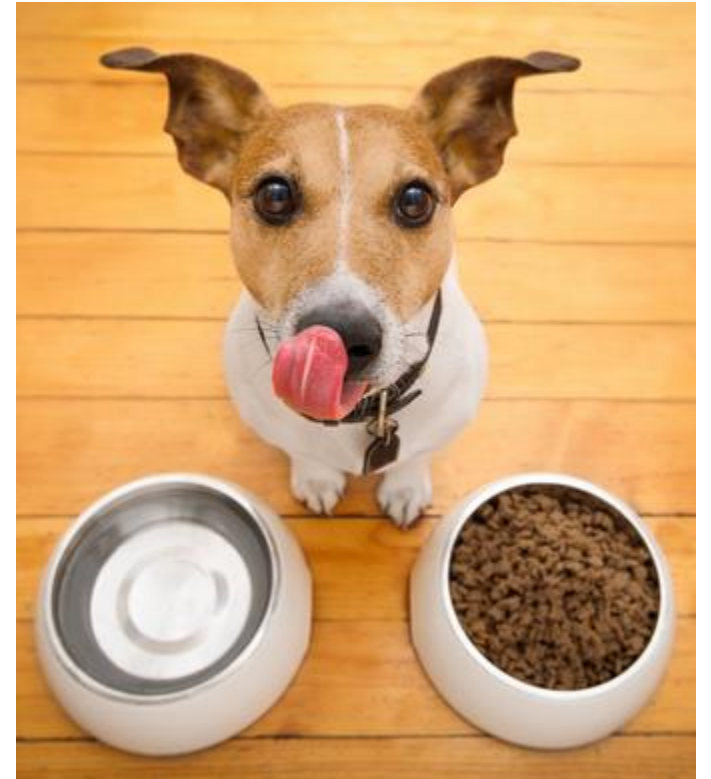
Overcome obstacles

- Executive support is critical
- PAM should be collaborative
- Build power users in the community



Lessons learned

- Senior leadership and stakeholder commitment are critical
- Realistic timeline, roadmap and scoping
- Careful design and testing
- Eat your own dog food
- Maturity models help keep you on track
- Automate everything



Key Takeaways

- Developing an effective PAM service is difficult but rewarding
- Maintain your roadmap
- Automate, automate, automate
- If You Fail to Plan, You Plan to Fail



Thank you!

Ken Robertson

- Mastodon - @ken@infosec.exchange
- Twitter - @krobert7
- Maturity model - https://gitlab.com/kenrobertson/PAM_maturity_model
- Divio system - <https://documentation.divio.com/>
- Automate!

