Remctl at the University of Michigan

Thomas L. Kula

Information Technology Central Services

University of Michigan

2008 AFS and Kerberos Best Practices Workshop

The Trouble With Delegation

- Many systems have two levels of authority: normal user, and super user

- I'm fine with that

The Trouble With Delegation

- My organization is more complex than the imagination of most system developers

- "We are the central IT organization for a University with no center"

  – (And hardly any organization....)

The Trouble With Delegation

- I know how to build tools to do what I want
- I just need a helper that allows me to define who can actually run that tool

Enter History

- We have historically used a home-grown tool
- Based on Rx
- Requires enough of a CM to get tokens
- We're the only people using it
- Calls a Perl module
- Getting users outside of my group usually involved some work

Enter remctl

- Started using in 2007 or so
- Used by people other than us
- Active development
- Not restricted to Perl
- GSSAPI/K5 based
- Usually not hard to compile on relatively modern systems

Example Uses

- Virus Busters Team releasing R/O AFS volumes containing their virus definition files at will

Example Uses

- LS&A College no longer had the resources to maintain their own AFS cell
- We were able to absorb their cell
- Give them dedicated fileservers
- Delegate authority to create, modify and delete any "lsa." AFS volume

Example Uses

- Delegated password changes
- Our Medical Center IT folks use a web frontend that calls remctl to change the passwords of a group of their users
- The MCIT backend guy got excited when we showed him how easy it was to use remctld himself

Other Potential Uses

- Since our Web SSO system (Cosign) allows CGIs to run with user-acquired credentials, we can easily fire off calls to remctl.

- Allows us to delegate administrative tasks to users

- Allows us to keep administrative credentials isolated on the remctl server

Use Outside Delegation

- Internally we frequently need to fire off one-shot commands to various systems

- Remctl allows us to leverage the pervasiveness of kerberos in our environment to do that

General Philosophy

- We can easily write tools to use administrative credentials to perform tasks

- Remctl makes it easy for us to call those tools, limiting who can call them, and verifying the identity of the caller

Finally

- A nifty, well-crafted tool that does one thing very well
- Our use is growing

Remctl at the University of Michigan

Thomas L. Kula

Information Technology Central Services

University of Michigan

tkula@umich.edu

2008 AFS and Kerberos Best Practices Workshop

http://kula.tproa.net/talks/afskbpw2008/kula-umich-remctl.pdf