Agenda

• Introduction
• Resource Intensity and Laziness
• Our Solution
• Demo Time
Who Are We?

Michael McAtee
- Manager
- Pentester
- SysAdmin
- [something funny here]

Lucas Morris
- Senior Manager
- “Manager”
- Pentester
- Code Monkey
Introduction

• Centralized Resources
  – In addition to purpose built tools, we now sometimes have purpose built hardware for them

• But how do we share this hardware?
  – Cracked ended (not paused)
  – No idea where the output is
  – Did they even ask?
Resources and Laziness

• To share, we may have to break some rules:
  – Shared Accounts
  – Shared GNU Screens
  – Manual Data Cleanup

• Did your co-worker use the right options?
  – Can you even resume their work when you’re done?
Resources and Laziness

• We also spent all this money:
  – What happens if the box is sitting idle?
  – How do we prove to management we’re using this expensive “toy”?
  – Are we using all of the resources (CPU in addition to GPU)?
Our Solution: CrackLord

- A distributed system of daemons to manage all of these resources and our queue
  - Input to Output are all modular and generic, anyone can write a plugin to wrap tools
  - Will manage jobs between multiple systems
  - Can use Cloud based systems (AWS!)
  - RESTful API that can be scripted into your workflow
Resources and Tools

• Resources are the server running on the hardware where the magic happens:
  – Can allow tools to use CPU, GPU, Network, or anything else.
  – Works on both Windows and Linux

• Tools are the Go packages that wrap industry tools.
  – Extensible, you can wrap / do just about anything
  – Tools define the resources they use, and can be configured on individual resources if desired
Queue and API

- The centralized point where everything is managed:
  - API is extensible, tools define what data is requested and what output looks like.
  - Can be reordered easily (drag and drop!)
  - Jobs automatically pause / resume
  - Uses TLS and cert based authentication between resources
  - The API and default GUI are designed to be tool generic, allowing the plugins to define input form and output.
CrackLord: Design

- Tools
- Queue
- Resources
- API (Users)
Demo Time
Black Hat Sound Bytes

• CrackLord is extensible to your needs

• Maximize your hardware investments

• BUILD PLUGINS!
Thank You!
https://github.com/jmmcatee/cracklord

• Michael McAtee
  ✉ => jmmcatee@gmail.com
  🐦 => @michaelmcatee
  🦁 => github.com/jmmcateee

• Lucas Morris
  ✉ => emperorcown@gmail.com
  🐦 => @lucasjmorris
  🦁 => github.com/emperorcow