CTF: State-of-the-Art and Building the Next Generation

ASE 2017

Clark Taylor
Lawrence Livermore National Laboratory
University of Arizona

August 15, 2017
Motivation

- Cyber Defenders Program
- 8 years of Cyber Defenders CTF
  - Different competitions from Sandia and LANL
  - On-site and remote setups
- 3 years of Cybercraft
  - Summer long for Cyber Defenders
  - Events at library, National Science Bowl
  - Based on PicoCTF framework
- Results?
  - Surveys indicate generally good results
General Competition Goals

- Ease of use
  - Alternative target audiences

- Keep costs down
  - Hardware
  - Administration

- Competition realism
  - Policy

- Variety of modes
  - Engaging for a range of skill levels

- Research/evaluation outcomes

- Framework extensibility
General Strategies for Cyber Defenders

- Hardware costs
  - Raspberry Pi and Kali Linux

- Add storylines and custom content for realism

- Design content for training (Cybercraft)

- Collect surveys and summary data
Can we do Better?

- Cyber Defender competitions had shortcomings
- What other CTFs are out there?
- Studied 39 different CTFs
- Found commonalities
  - Framework vs monolithic
    - Open source?
  - Dynamic vs static challenges
    - “Challenges” refers to individual puzzles or tasks for which points are awarded
  - Policy topics
State of the Art

Current Competitions

- Static
- Dynamic
- Open Source
- Policy
- Total

[Bar chart showing distribution across categories]
Shortcomings

- Ease of use
- Realism
  - Challenge types
  - Policy integration
- Training oriented modes
- Data collection
- Content development ease
Proposed Framework: Catalyst

- Introducing the Catalyst Security Challenge (CSC)
- Aims to solve these problems
- Currently in progress
  - Building component-by-component
Ease of Use

- Entirely web based
  - Hosted on a LAN, with network management (including VPN) on game server
  - Plaintext-and-buttons configurable

- Automated provisioning
  - Provision the game server from the internet
  - Provision other components from the game server
    - Includes provisioning for participant terminals
    - Components can be virtual
    - Components can be inexpensive hardware

- GUI managed

- Target audience: Lowest common denominator
Realism

- Support for static and dynamic content
  - HTTP-based extensible grading system API

- Highly configurable
  - Plaintext configuration includes challenge text and parameter configuration

- Policy challenges
  - Built-in support for policy-oriented types of challenges
  - Additional realism and configurability better enables policy challenges
Training Modes

- Administrators can choose desired mode of competition
- Training modes disable features such as public scoreboards
- Support for challenge hints
- Challenge components may behave differently
  - Content developers can access the current mode for their components via the grading API
Data Collection

- Keylogger-on-steroids approach
  - Monitors endpoints
  - Installed automatically via provisioning

- Data curated on game server

- Visualization and filtering

- Goals:
  - Find novel approaches to solving challenges
  - Determine best practices and strategies
  - Evaluate efficacy of CTF for training/education
  - Evaluate participants
Data Collection: Current Visualization
Content Development

- No particular design patterns
  - Just has to be compatible with host OS and support HTTP
- Game server provisioning launches software via OS
- All communication done via grading API
Architecture
Questions?