Contest: Cyber Operations Exercise with Cryptol and SAW

Purpose and Description

CDEST is a team competition that is intended to enhance the Cyber Operations (CO) community by reducing time and cost of apprenticeship training and by improving the motivation and quality of the CO workforce. Critical to improving quality and reducing time and cost of apprenticeship training is ensuring a greater depth of knowledge and utility of CO skills in personnel entering the workforce. CO specialists 1) protect data, networks, and net-centric capabilities; and 2) neutralize the digital capabilities of adversaries including their ability to communicate, initiate attacks, and control infrastructure. Critical CO skills therefore include low-level programming, operating system weaknesses and vulnerabilities, network protocols, encryption, authentication, integrity protection, reverse engineering, and forensic analysis of traffic and data.

The CDEST competition can exercise some or all of those skills. CDEST supports a variety of Cyber Defense Exercises and consists of two major components: a modified operating system (OS) and scoring software (Scorer). The OS is distributed by a Contest Administrator (CA) to Teams some number of days, determined by the CA, before a contest commences. Teams examine the OS for configuration errors, protocol weaknesses, and any vulnerabilities the OS may have and attempt to mitigate or fix such problems. At contest start the Teams put the OSes online to provide a collection of internet services that are specified by the CA in advance. Also at contest start the CA initiates the Scorer: at approximately 1 minute intervals, but depending on the number of Teams participating, the Scorer checks to see if the services of a Team's OS are up. The Scorer adds 1 point to a Team's cumulative score for each service that is deemed working when a check is made. A real-time scoreboard is maintained during the contest. The scoreboard is generally publicly accessible although it can be made private. At contest end the team with the highest cumulative score wins. More importantly, an assessment is made as to how well a Team was able to defend its OS.

There are several ways CDEST can be used. The supplied OS may be replaced by another at the discretion of the CA. The services checked may be changed by modifying the CheckServices.java file and recompiling the Scoring software. The time between checks may be changed in the same way. Teams may be allowed to attack the OSes of other Teams in a variety of ways as well as defend their OSes from attack. The CA may require Teams to defend only and enlist other Teams, having seen the OS in advance, to attack the OSes. Finally, the CA may ask Teams to choose their own OS and require that a collection of services must

be running on particular ports.

Not included with CDEST are the many publicly available tools for analyzing, modifying, and controlling network traffic, for example firewalls. Teams should use those tools to anticipate and mitigate attacks. The CA will use those tools to analyze the performance of Teams after the contest is ended. CDEST is best used at the end of a course on cyber defense. It helps if Team members have also taken a course in vulnerabilities analysis. All labs in this module make use of a Linux shell. The Linux tutorial, presented as the first activity in this module, aims to familiarize a learner with shell commands and their actions.

Audience This module is intended for:

1 The general public
2 K-12 and college classes on cyber defense
3 people interested in cyber games

Objectives After completing the module:

1 Proficiency in using the Linux shell
2 Proficiency in OS configuration management
3 Knowledge of Reverse Engineering binaries with the aim of understanding how functions in the binary operate and whether there is malware or vulnerabilities in it
4 Knowledge of network protocols and vulnerabilities with the aim of understanding how to protect OS services from misuse

Keywords shell, tutorial, Linux, Ubuntu, commands, tools, cyber defense, cyber operations, configuration management, reverse engineering, operating system

Category cybersecurity > education

Delivery java applets and written documentation in pdf format

Team John Franco and Ethan Link

Assessment The applets provide the means for experimentation. Questions are asked in the documentation that help with the set up of experiments. The ideas that learners come up with is evidence that the module was successful.

Workflow No particular schedule was established

Environment All materials are contained in a single jar file. The jar file can be run on any computer where java version 14 or higher and some pdf reader such as acroread or evince are available. The jar file may be executed in the cyber range or learners may download the jar file (which is considered to be an executable file) and run it on their personal computers.