Carnegie Mellon University
Software Engineering Institute

Crucible

An Open-Source Application Framework for Cyber Training, Experimentation, and Exercise

Introducing Crucible

CRUCIBLE is an open-source application framework for creating and managing virtual environments and events.

Key Features

- Open-source application framework built on Angular and .NET Core software frameworks
- Modular design - extensive application programming interfaces
- Customizable, immersive, browser-based user interface
- Flexible integration of powerful, third-party, open-source tools
- Scenario-based cyber experimentation, exercises, and challenges
- Model topologies, simulate user activity, script scenario events
- Efficiency through automation
- Interoperability through open standards
- Options for building cyber terrain:
  - "Infrastructure-as-code" for scalability, iteration, and reuse
  - Form-based configuration for simple and quick

A Proven Track Record

Since 2018, Crucible has enabled large-scale United States (US) Department of Defense (DoD) cyber exercises, the President's Cybersecurity Challenge Competition, and partner nation information sharing and training initiatives.

Addressing Persistent Challenges

Crucible confronts challenges faced by platform administrators and content developers:

- manual configurations lead to high-labor costs and excessive human error—limiting scalability and automation
- proprietary range software leads to vendor lock-in and higher costs

Crucible Platform Features

Crucible platforms universally leverage OpenID Connect for authentication—and can be tailored for learning management, threat information sharing, and collaboration. Crucible platforms can also be focused on individual training and/or team exercising.

Open-Source Platform Integrations:

- **Keycloak**, an open-source identity authentication service that implements the OpenID Connect authentication protocol.
- **Moodle/H5P**, an interactive learning management system, enables embedding interactive quiz content and recording user-experience data to a learning record store using the Experience API (xAPI).
- **MISP**, threat information sharing platform for sharing, storing and correlating Indicators of Compromise.
- **NextCloud Hub**, a self-hosted solution that integrates file storage, chat/talk, email, calendar, office and more in one federated content collaboration platform.

INDIVIDUAL TRAINING

Crucible can be a platform for individual practice or competition. These platforms tend to feature the following two applications:

Going Simple: Labs/Challenges

Crucible's **TopoMojo** application enables design of simple labs and challenges using form-based configurations. Select and configure virtual machines, define networks, and write a guide.

Novice Crucible content developers can easily get productive by using TopoMojo. Choose this app when the benefits of more advanced "infrastructure as code" automation are not needed. TopoMojo supports the configuration and deployment of small virtual environments to two types of hypervisors: VMware vSphere ESXi and Proxmox Virtual Environment KVM (open source).

Crafting a Challenge Competition

Crucible's **Gameboard** application provides game design capabilities and a competition-ready user interface for running your own cybersecurity game.

A Crucible content developer can create, clone, manage, and delete games and challenges—for competition or practice.

TEAM EXERCISING

Crucible can also support more advanced needs commonly found within concept experimentation and team exercising (table-top, functional, and full)—using some of the following ten applications:

Designing User Experiences

Crucible's **Player** application is the exerciser's window into the virtual environment. Player enables assignment of team membership as well as customization of a responsive, browser-based user interfaces using various integrated applications. A Crucible content developer can shape how scenario information, assessments, and virtual environments are presented through the use of integrated applications.

Open-Source Integrations:

- **osTicket**, a support ticket system, manages cyber range service requests.
- **Mattermost**, **Rocketchat**, **Nextcloud Talk** chat services.
- **Stalwart**, **Roundcube**, web-based email service.

Coding a Topology

Crucible's **Caster** application enables coding design and deployment of a cyber topology. With Caster Designs, an intermediate content developer can avoid scripting Terraform code and simply define variables within pre-configured Terraform modules.

Caster supports the design and deployment of virtual environments to a variety of hypervisors: VMware vSphere ESXi, Microsoft Azure HyperV (public-cloud), Amazon Web Services Xen/Nitro (public-cloud), and Proxmox Virtual Environment KVM (open source).

Open-Source Integrations:

- **Terraform/OpenTofu**, an "infrastructure-as-code" tool, enables scripted deployment of cyber infrastructure.
- **GitLab**, a version control system and code-repository, is used to store Terraform/OpenTofu modules.

Crafting a Scenario

Crucible's **Blueprint** application enables the collaborative creation and visualization of a master scenario event list (MSEL) for an exercise. Scenario events are mapped to simulation objectives.

Crucible's **Steamfitter** application enables the organization and execution of tasks on virtual machines.

Open-Source Integrations:

- **StackStorm**, an event-driven automation platform, scripts scenario events and senses the virtual environment.
- **Ansible**, a software provisioning, configuration management and application deployment tool, enables post-deployment provisioning of services to infrastructure.

Modeling the Internet

Crucible's **Greybox** virtual machine provides the illusion of connectivity to the real Internet: a realistic BGP backbone topology with point-to-point link delays based on physical distance between the routers' real-world locations, combined with application services (HTTP, DNS, email, etc.).

Open-Source Integrations:

- **CORE (Common Open Research Emulator),** a tool for building virtual networks that run in real time.

Animating Activity

Crucible's **GHOSTS** Non-Player Character (NPC) automation and orchestration framework deploys and shapes the activities of NPCs using GenAI.

Open-Source Integrations:

- **Ollama**, a platform designed to run llama, mistral, and other open source large language models locally.

Evaluating Threats

Crucible's **Collaborative Incident Threat Evaluator (CITE)** application enables participants from different organizations to evaluate, score, and comment on cyber incidents. CITE's situational awareness dashboard allows teams to track internal actions and roles.

Displaying Incident Information

Crucible's **Gallery** application enables participants to review cyber incident information based on source type (intelligence, reporting, orders, news, social media, telephone, email) categorized by critical infrastructure sector or any other organization.

Assessing Performance

Crucible's **SEER** application enables assessment of team performance. During events, participants are challenged to perform mission-essential tasks and individual qualification requirements. Map performance assessments to training objectives to scenario events.

Launching an On-Demand Exercise

Crucible's **Alloy** application enables users to launch an on-demand event or join an instance of an already-running exercise. Following an event, Alloy can also provide a summary of knowledge and performance assessments.

Operational Deployment

Crucible applications are deployed as **Docker** containers, which employ operating system level virtualization to isolate containers from each other. Container deployment, scaling, and management services are obtained using **Kubernetes**, a popular container-orchestration system. Kubernetes workflow and cluster management are performed using **Argo CD**, a popular open-source GitOps toolset. The SEI owns and operates an on-premises instance of Crucible that can deploy virtual environments to VMware, Proxmox, or to a cloud provider:

Fortress
fortress.sei.cmu.edu

Learn More

See the full documentation at **cmu-sei.github.io/crucible/** and **cmu-sei.github.io/GHOSTS/**.

For more information, email **info@sei.cmu.edu**.

About the SEI

Always focused on the future, the Software Engineering Institute (SEI) advances software as a strategic advantage for national security. We lead research and direct transition of software engineering, cybersecurity, and artificial intelligence technologies at the intersection of academia, industry, and government. We serve the nation as a federally funded research and development center (FFRDC) sponsored by the U.S. Department of Defense (DoD) and are based at Carnegie Mellon University, a global research university annually rated among the best for its programs in computer science and engineering.

Contact Us

CARNEGIE MELLON UNIVERSITY
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE; PITTSBURGH, PA 15213-2612

sei.cmu.edu
412.268.5800 | 888.201.4479
info@sei.cmu.edu

Copyright 2024 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific entity, product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute nor of Carnegie Mellon University - Software Engineering Institute by any such named or represented entity.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License. Requests for permission for non-licensed uses should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

CERT® and Carnegie Mellon® are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM24-1640