Spring Cyber Defense Competition 2016

Scenario

IOWA STATE UNIVERSITY, INFORMATION ASSURANCE CENTER
2016

Version 1.0

Cluster Deployment Company

Internal Memo

Welcome!

Welcome to the Cluster Deployment Company (CDC). Here at CDC, we offer enterprising developers a way to run their code on our custom CodeRunner software stack. We pride ourselves on our security and our commitment to Open Source. Our infrastructure features top-notch security and encryption powered by RSA and SSH. We pride ourselves on security, and hope that your team will be a welcome addition to our family and a valuable asset to CDC.

Client Overview

Our CodeRunner infrastructure uses two custom tools: keyescrow and crconsole. The keyescrow utility is used to retrieve SSH keys for our users. These keys are used to access the environment as well as by the CodeRunner console (crconsole). Our website's Client Area contains a frontend to these tools as well.

Signing Up

You can sign up for a client account on our website. Creating an account on the website gives you access to the Client Area, as well as our GitLab instance and the Shell hop to the CodeRunner infrastructure.

Uploading Code

The first thing you will want to do once you have a CDC Client Account is retrieve your SSH Public Key through the Client Area and add it to your GitLab account. You can find instructions for adding GitLab keys in any GitLab documentation. Once you have added your key to GitLab, you can use your Private Key (also retrieved through the Client Area) to push code to GitLab.

Running Code

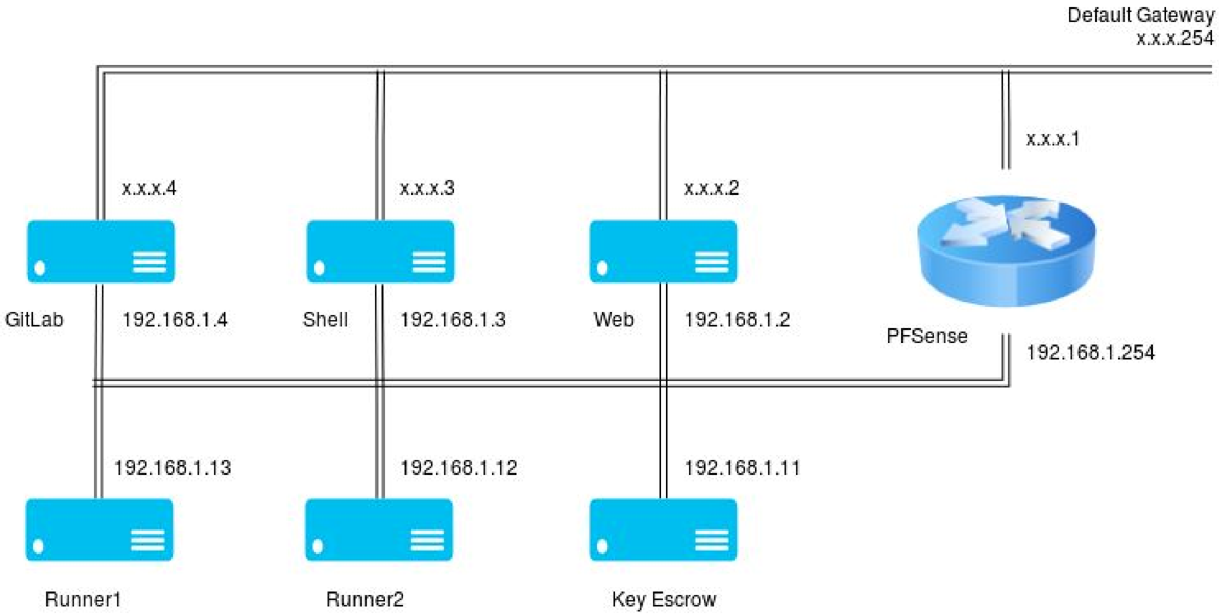
The easiest way to run your code is through the client area. Use the provided options to deploy a project (repository name) to one of our available runner servers. You can then build and run the code, as well as view the output of the job.

You can also connect to our Shell Hop using your Private Key. Once you are on the shell hop, you can use crconsole to deploy and run your code. The command-line tools offer additional flexibility over the web interface.

Note: Before you are able to use crconsole, you must retrieve your private key on the Shell Hop using the keyescrow utility.

Network Diagram

This diagram shows our current network layout. Everything is already working and stable in this configuration. You can keep it as it is, or use this as a starting point for a completely different setup. Take as much downtime as you need to move things over; our focus is on having a secure and efficient network.



Servers

The servers listed below have been provided to your team and have various access requirements that must be met by your team. While you may make major configuration changes for the sake of increasing security or usability, your servers must provide all required and original functionality.

Web Server (www.teamN.isucdc.com)

username: cdc
password: cdc

This Linux web server runs our custom corporate website, which is written in PHP. It provides everything current and prospective clients need to create an account and begin running their code on our CodeRunner infrastructure.

Access Requirements

- Must be accessible from the Competition Network
- Must have access to the Runner servers and Key Escrow servers

- New Clients must be able to create accounts and have access to the Client areas to run code and perform other Client actions
- New Clients should should immediately have access to gitlab via the web application and ssh access to the shell hop using their ssh keys.
- Administrators must have SSH access with unrestricted sudo privileges on the box
- **Users** must be able to view and update their account information and retrieve their SSH keys
- **Admins** must be able to view and modify the account details of all users

Services

- SSH (Port 22) - Administrators
- HTTP(S) (Port 80 or 443)

Flags

- Credit Card information for "cdc" user (Blue Flag)
- /etc/ (Blue Flag)
- /root/ (Red Flag)

GitLab (git.teamN.isucdc.com)

username: cdc
password: cdc

All of our code, as well as all client code, is hosted on a public GitLab server. Our company prides itself on our commitment to Open Source.

Required Access

- Must be accessible from the Competition Network
- Runner servers and Shell server must be able to access GitLab using the crconsole utility
- Clients must be able to log in to the GitLab service with an account created on the main web site
- Clients must have full GitLab access for creating and managing repositories
- Clients must be able to use Git to push or fetch code from Git
- Administrators must be Admin-level access to the GitLab web application
- Administrators must be able to access the server over SSH from the Competition Network
- Administrators must have unrestricted sudo privileges on the box

Services

- HTTP(S) (Port 80 or 443)
- SSH (Port 22) for Administrators and Git

Flags

- /root/ (Red Flag)

- /etc/ (Blue Flag)
- Push to GitLab repo /cdc/flag (Red Flag)

Shell Server (shell.teamN.isucdc.com)

username: cdc
password: cdc

This server allows clients to have a Linux shell environment to test code and submit jobs to our CodeRunner infrastructure. This functionality duplicates some of the Client Area functionality in the main web site, but provides a different interface.

Required Access

- Must be accessible from the Competition Network
- Must have access to GitLab, Key Escrow, and both Runner servers
- Clients must be able to access the box over SSH with keys obtained from the Client Area in the main website
- Clients must have read-write access to their home directory and /tmp
- Clients must have access to development tools (gcc, g++, dmd, ruby, python, javac, make, cmake, go)
- Administrators must have unrestricted sudo privileges on the box

Services

- SSH (Port 22) for Administrators and Clients

Flags

- /root/ (Blue Flag)

Key Escrow Server (Internal)

username: cdc
password: cdc

This server stores the SSH keys used by Clients to access the Shell box and CodeRunner infrastructure. It uses a custom protocol and toolset to deliver keys to the users as well as configure authorization on the destination server.

Required Access

- Clients must be able to retrieve their public and private keys using the Client Area of the website or the key escrow software on the Shell server
- Clients must be able to dispatch public keys to the Shell server from the Client Area of the website
- Administrators must have SSH access via pivot from Shell, GitLab, or WWW

- Administrators must have unrestricted sudo privileges

Flags

- /etc/ (Blue Flag)
- /root/ (Red Flag)

Runner 1 (Internal)

username: cdc
password: cdc
This is a Fedora server supported by CodeRunner.

Required Access

- Clients must be able to run code on Runner1 using the CodeRunner tools on the Shell server or through the website
- Administrators must be able to access it over SSH by pivoting through Shell, GitLab, or WWW
- Administrators must have unrestricted sudo privileges
- Administrators must be able to view the CodeRunner logs
- Must have the same set of development tools as the shell server

Flags

- /root/ (Blue Flag)

Runner 2 (Internal)

username: cdc
password: cdc
This is a Debian server supported by CodeRunner.

Required Access

- Clients must be able to run code on Runner2 using the CodeRunner tools on the Shell server or through the website
- Administrators must be able to access it over SSH by pivoting through Shell, GitLab, or WWW
- Administrators must have unrestricted sudo privileges
- Administrators must be able to view the CodeRunner logs
- Must have the same set of development tools as the shell server

Flags

- /root/ (Blue Flag)

PFSense

username: admin
password: cdc
This server is provided, but is not required by the scenario.

Notes

Flags

This scenario includes two types of flags. Blue Flags must be placed by you onto your server prior to the beginning of the attack phase. These Blue Flags can be files, in which case the flag file must be placed in the given directory. These flags can be protected but must have realistic permissions for the directory they are in. Blue Flags can also be database entries, in which case the table, column, and row for the flag will be detailed by the scenario. The table for the flag will be described in terms of the application which uses the table, not the server which hosts the database. Red flags are planted by Red Team if they are able to gain write access to the appropriate directory (usually requiring superuser access).

In this scenario, Blue Flags placed in the */etc/* directory must have the permissions *r--r--r-- (aka 444)*.

Users

As usual, required users are outlined in the Users and Accounts document. Additionally, any visitor to the website must be able to sign up and create a Client account. Administrators must have unrestricted sudo access on all servers and global administrator access on GitLab as well as the ability to view all client details in the web application. Client users must be able to access their own GitLab accounts and the Client Area of the website. Clients must also be able to obtain shell access to the shell hop using their SSH Private Key. Client users must have access to development tools on the shell hop, and must be able to use the crconsole utility's deploy, build, run, stdin, stdout, and stderr functions to deploy GitLab repositories and execute code on all runner servers.

In this competition, the “cdc” user *must* have a client account on all of the systems that clients are required access to. However, you may set the password to anything of your choice.

Migration

For this scenario, Blue Teams are permitted to migrate/rebuild systems. You are free to replace any of the provided servers or applications with your own, with respect to the usual competition rules. Any operating system must be rebuilt using the same distribution, but a different version can be used. Custom software can be rewritten in any language/framework. The user interfaces of the custom utilities (keyescrow and crconsole) must maintain compatibility, including exact syntax for all documented functionality. All documented web functionality must remain available.

Additional Servers

While not required, you are permitted to build additional systems (i.e. IDS/IPS, AD, firewalls) to augment your systems. Remember to consult the “Remote Setup” document, as there are some important guidelines you must follow when creating new VMs.