Exploiting AWS ECR and ECS with the Cloud Container Attack Tool (CCAT)

By Jack Ganbold

Published: 2019-08-27 · Archived: 2026-04-05 21:17:23 UTC

UPDATE: As of 10/03/19, CCAT now supports Container Registry on GCP!

Introduction

Docker and other container technologies are becoming increasingly popular and are being adopted by many companies. In recent cloud pentesting engagements, we have similarly noticed that many of our clients use container technology to run their systems. Although there has been research and tool development on containers and their security, most of those are focused on image analysis and finding known vulnerabilities.

Due to this lack of tools, we decided to build one for ourselves and named it the Cloud Container Attack Tool (CCAT for short). CCAT is different in that it utilizes containers for exploitation in the cloud through backdoors and malicious Docker images. In this post, we will dive into what this tool does and how to use it to leverage Docker for attacks against AWS ECS and ECR.

Docker Containers on AWS

AWS supports running Docker in order to provide users with a “highly reliable, low-cost way to build, ship, and run distributed applications.” For those unfamiliar, the following are a few different services built for working with containers in AWS.

Amazon ECS, EKS, and ECR

One method of running containers on AWS is through Amazon Elastic Container Service (ECS). Amazon ECS is “a highly scalable, high-performance container management service.”

Amazon Elastic Container Service for Kubernetes (EKS) is another service that can be used to run containers on AWS. It allows you to “deploy, manage, and scale containerized applications using Kubernetes on AWS.”

Amazon Elastic Container Registry (ECR) is a container repository used to store Docker images. The images are encrypted and compressed at rest so that they are quick to pull and secure.

Both Amazon ECS and EKS can pull Docker images directly from Amazon ECR when deploying containers. Through this, we can use backdoored containers to compromise massive environments with ease.

The Cloud Container Attack Tool

We created the Cloud Container Attack Tool (CCAT) for testing the security of cloud container environments. Currently, CCAT *(pronounced "sea cat")* is only compatible with AWS, however, we are working on expanding it to support other cloud vendors and adding more exciting features.

You can find [CCAT on our GitHub here](#).

How to Install CCAT

Below are the prerequisites for installing CCAT:

- Python 3.5+ is required.
- Docker is required. Note: CCAT is tested with the Docker Engine 19.03.1 version.
- AWS named profile is required.

Once you have all of the prerequisites, there are a few different ways to install CCAT–from source code or using CCAT's Docker image.

Installing from source code:

```
git clone https://github.com/RhinoSecurityLabs/ccat.git
cd ccat
python3 setup.py install
python3 ccat.py
```

Installing using CCAT's Docker image:

```
docker run -it -v ~/.aws:/root/.aws/ -v /var/run/docker.sock:/var/run/docker.sock -v ${PWD}:/app/ rh
```

-v ~/.aws:/root/.aws/

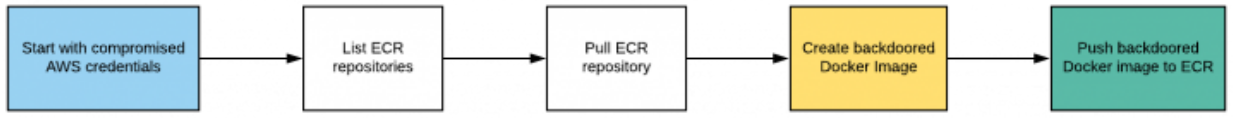
This argument mounts your local AWS configuration files into the Docker container when it is launched. This means that any user with access to the container will have access to your host computer's AWS CLI credentials.

-v /var/run/docker.sock:/var/run/docker.sock

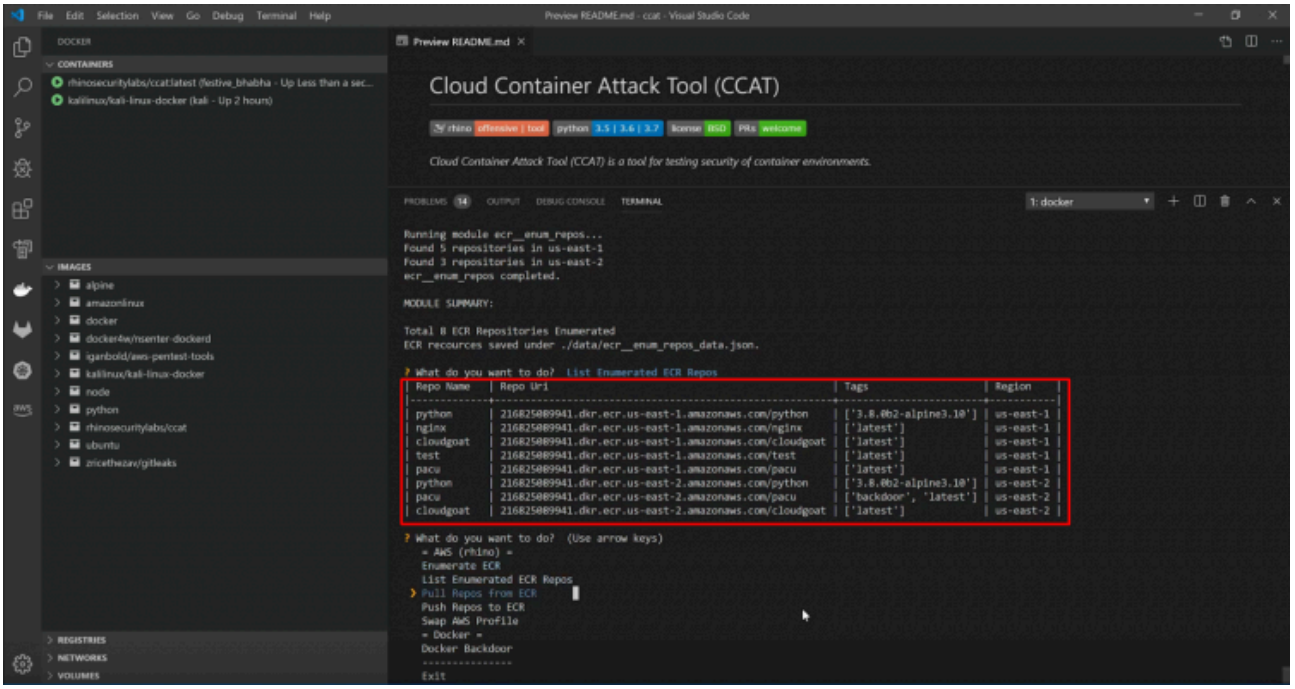
This argument mounts your local Unix socket that Docker daemon listens on by default into the Docker container when it is launched. This means that users with access to the container will have access to your Docker daemon, meaning they could escape to your host computer with ease.

Exploitation Walkthrough with CCAT

In order to demonstrate how to use CCAT, we will run through a small example scenario below, where an attacker uses CCAT to abuse compromised AWS credentials for further exploitation in the AWS environment.

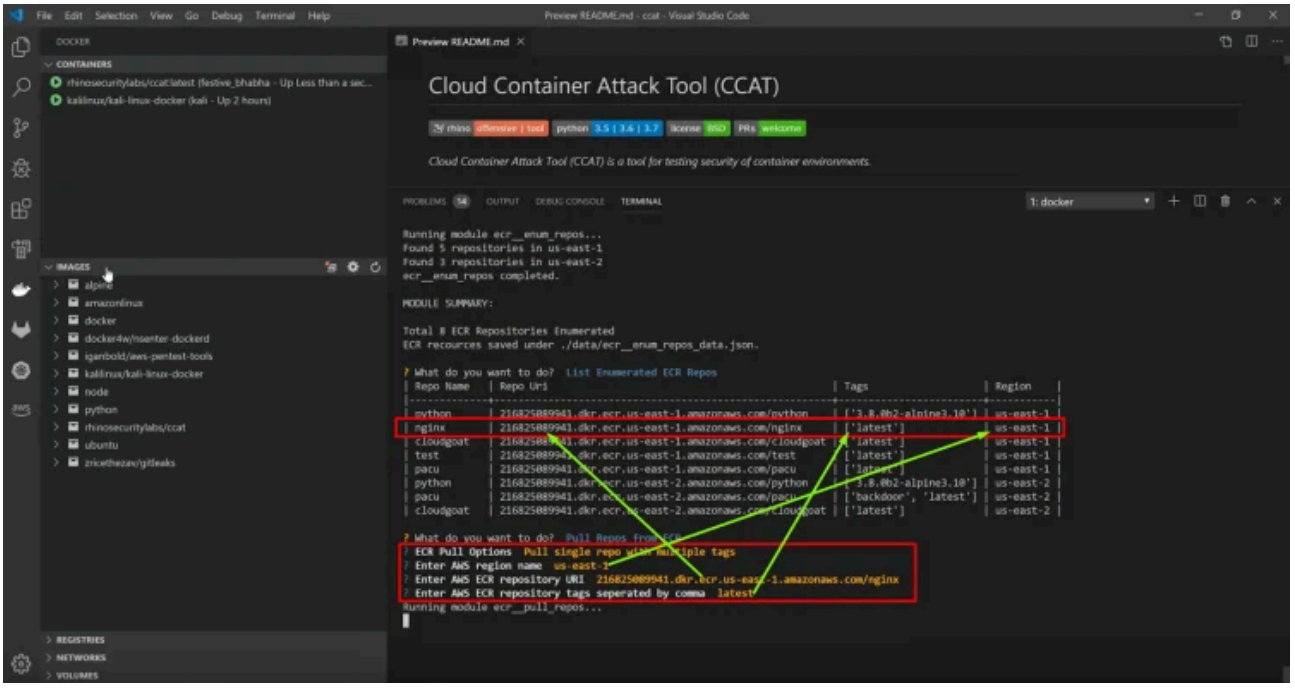


1. The attacker explores the AWS environment and discovers they are able to list ECR repositories using compromised AWS credentials.



Using the "Enumerate ECR" module to collect information about ECR repositories and list the collected repositories

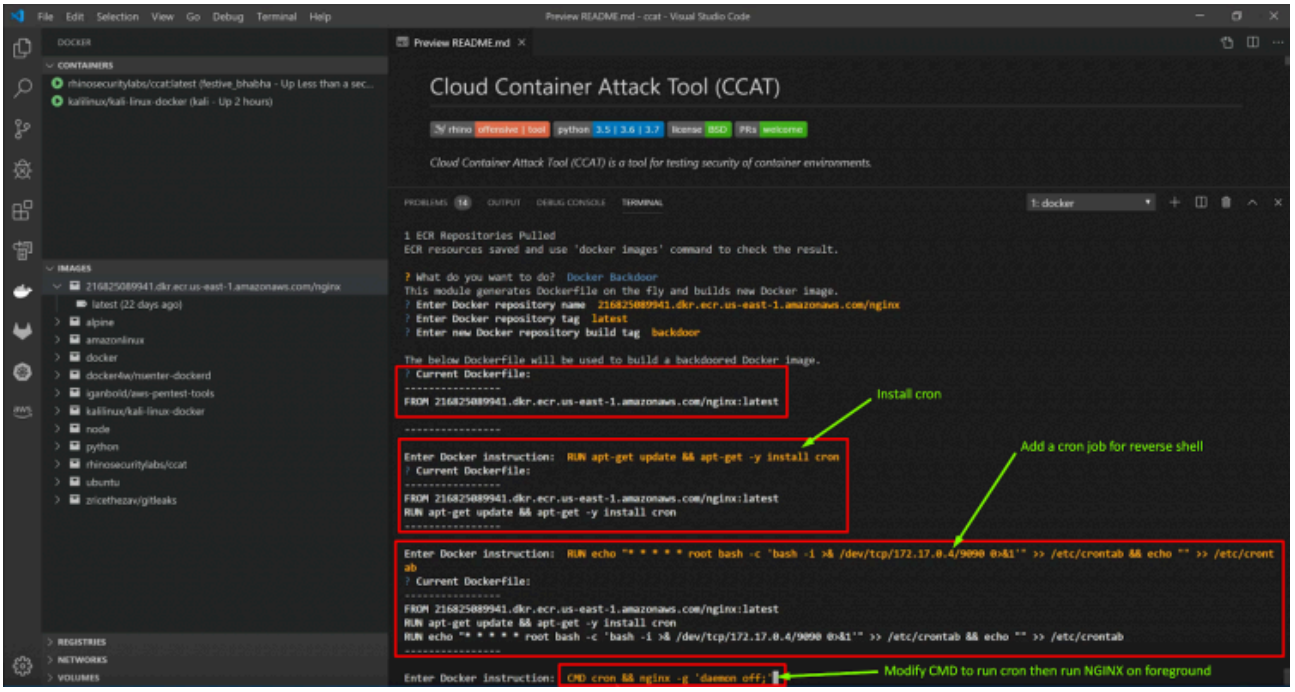
2. The attacker finds that their target uses an NGINX Docker image and pulls that Docker image from ECR.



Using the “Pull Repos from ECR” module to download the target ECR repository

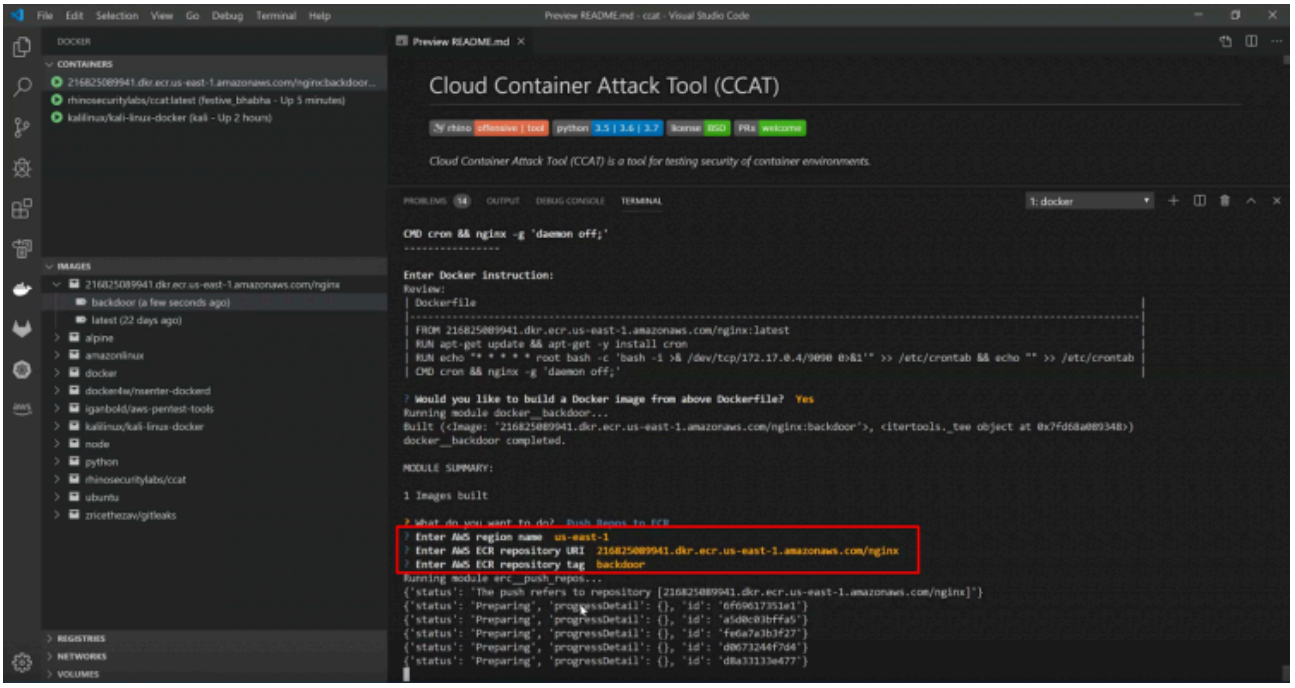
3. The attacker decides to create a reverse shell backdoor in the pulled NGINX Docker image.

This module generates a Dockerfile on the fly and builds new a Docker image from that file.



Using the “Docker Backdoor” module to generate a Dockerfile, add the reverse shell configuration, and overwrite the default CMD command

4. Finally, the attacker pushes the backdoored Docker image to ECR.



Using the “Push Repos to ECR” module to push a backdoored image

If you encounter any issues with the installation or usage of CCAT, please open an issue on the GitHub page.

Conclusion

We built CCAT to help the community better understand the security implications of container-based services, especially due to containers’ increasing popularity and the lack of offensive tools in the space. We are actively working to add more exciting container exploitation features and to support multi-cloud vendors.

For defensive purposes, you can use the Docker Bench for Security scripts to check common best practices for deploying Docker containers in production. For Kubernetes, you can also use Kube-hunter to hunt for security issues in your Kubernetes clusters.

If you’d like to talk AWS security or get some help with CCAT from the developers, join our community Slack team: the Pacu/CloudGoat/CCAT community Slack.
