Covenant C2 Framework: Practical Guide

redfoxsec.com/blog/power-of-covenant-c2-framework-a-comprehensive

Kunal Kumar July 22, 2023

Power of Covenant C2 Framework

- July 22, 2023
- Red Team
- Kunal Kumar

In the ever-evolving world of cybersecurity, staying one step ahead of malicious actors is crucial. Command and control (C2) frameworks play a vital role in post-exploitation activities, allowing security professionals to execute payloads on compromised hosts and gain control over the target network. One such framework that has gained significant attention is Covenant. In this comprehensive guide, we will explore the power of the Covenant C2 framework, from installation to executing advanced tasks, enabling security experts to bolster their defensive strategies.

Section 1: Getting Started with Covenant

Before we delve into the functionalities of Covenant, let’s start by understanding how to install and set up this powerful C2 framework.

Step 1: Installation

To begin, ensure that you have a host machine capable of running Covenant. While a Windows 10 workstation is suitable, any recent Windows Server OS or Linux distribution supporting .NET Core will suffice. The following applications must be installed on the host machine:

- Git
- .NET Core

Once the prerequisites are in place, proceed with the installation process:

1. **Clone the Covenant project repository using the following command:**

```
git clone --recurse-submodules https://github.com/cobbr/Covenant
```

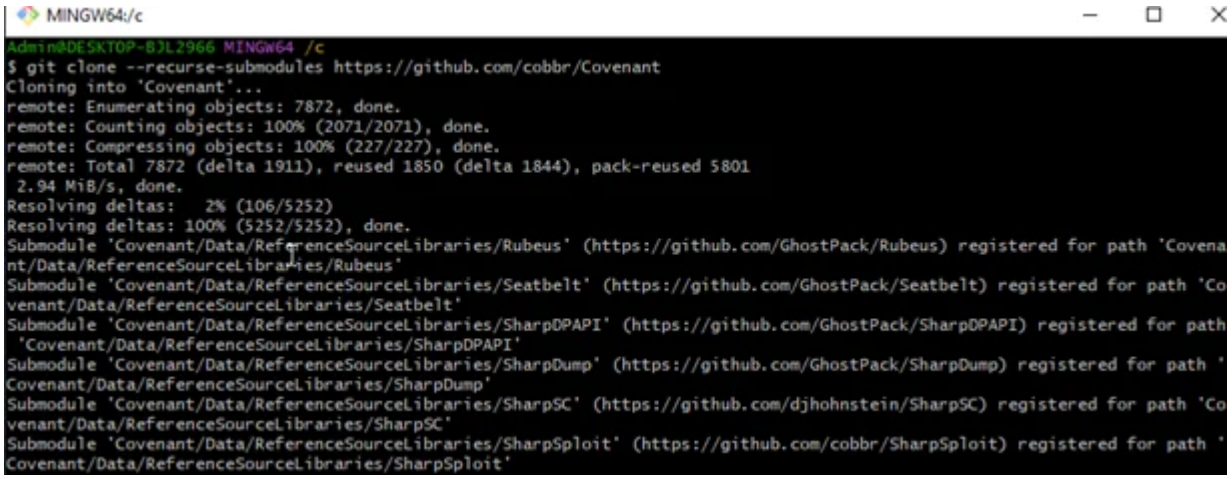

Figure 1: Downloading repository

Note: Before installing Covenant, we need to set defender Exclusions or Turn off windows defender. Here am adding exclusion to Covenant folder.

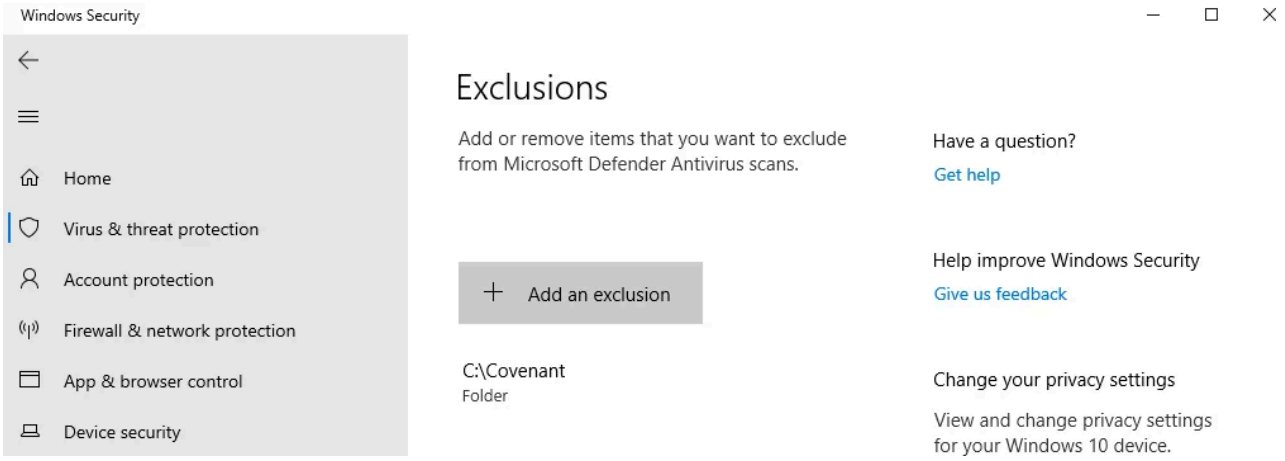

Figure 2: Adding Exclusions in defender

2. **Navigate to the Covenant directory:**

```
cd Covenant/Covenant
```

Figure 3: cd Covenant/Covenant

3. Build the application:

```
dotnet build
```

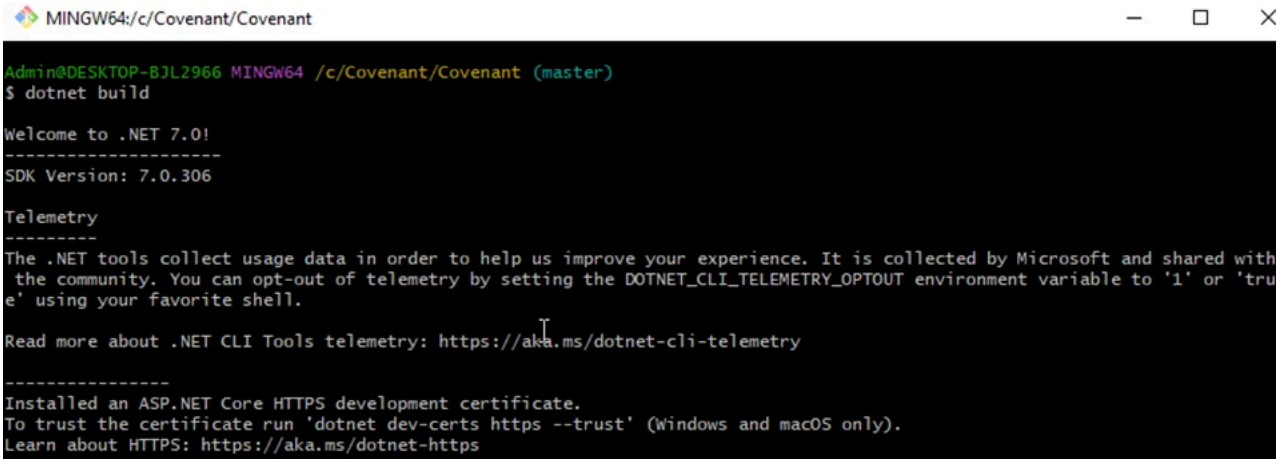

Figure 4: Building Application

4. Run Covenant:

```
dotnet run
```

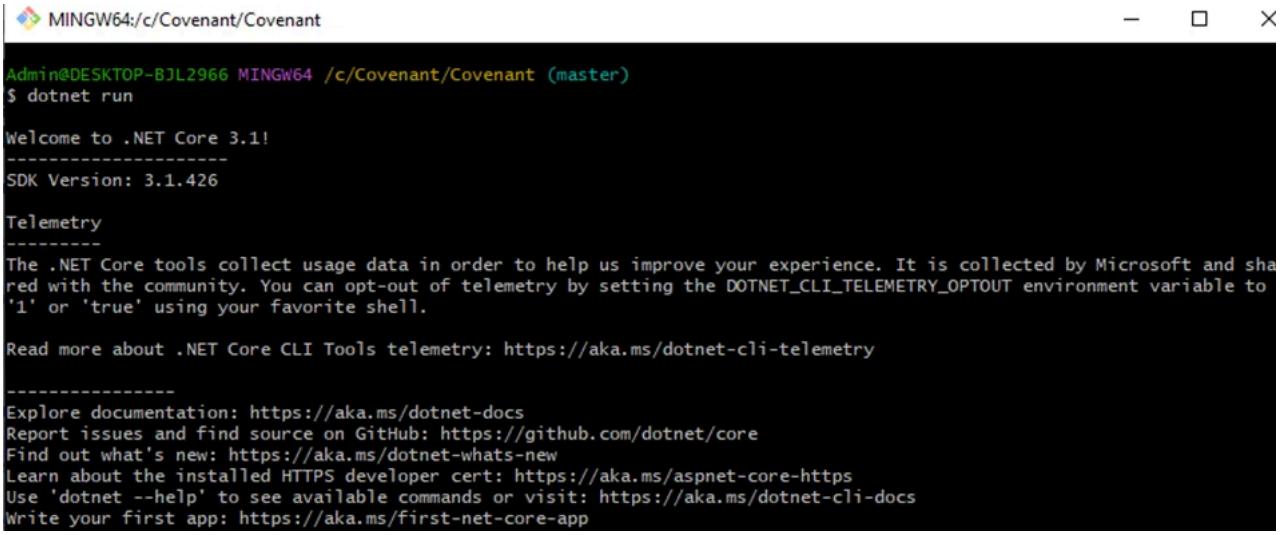

Figure 5: Running Covenant

Alternatively, you can run Covenant in a Docker container for added convenience.

Step 2: Creating an Account

With Covenant up and running, access the Covenant application interface on its default web port of 7443. You will be prompted to set up a user account upon initial access. Follow the registration process, and once completed, you will be redirected to the User Management page, where you can explore the various capabilities of Covenant.

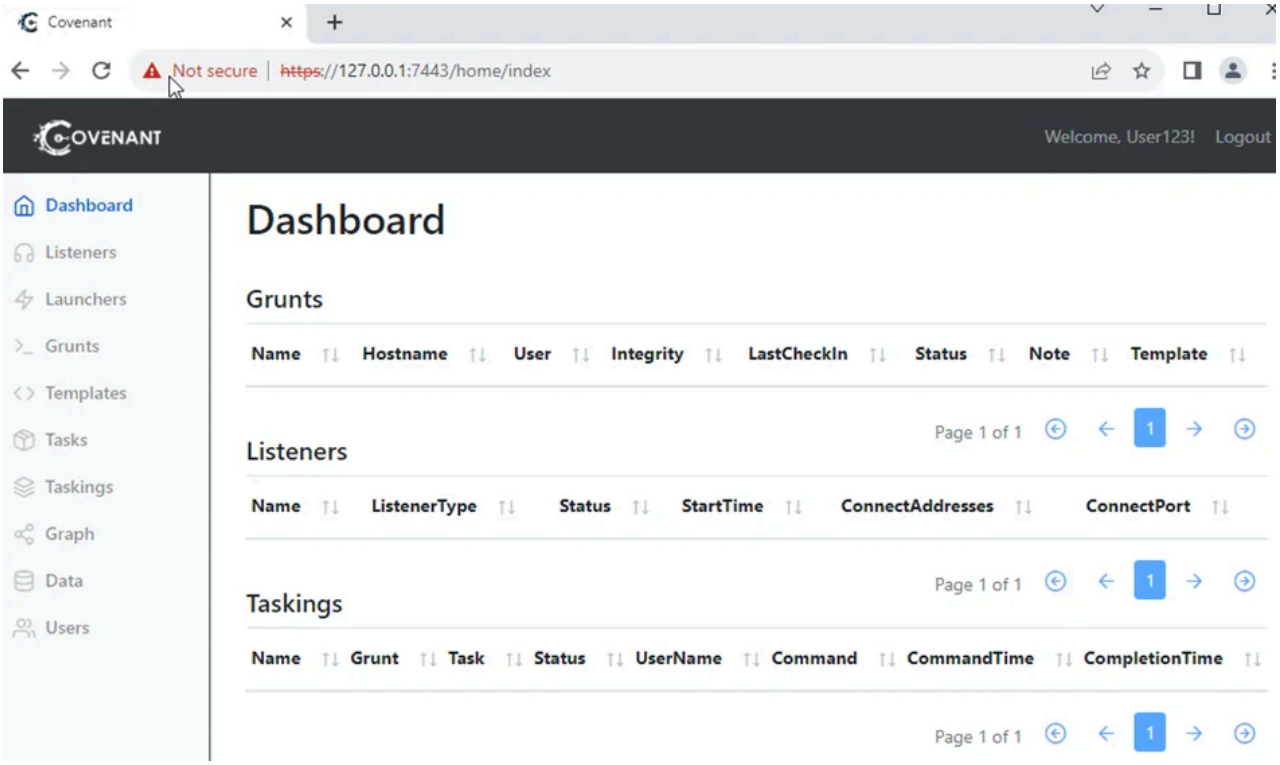

Figure 6: Dashboard

Section 2: Configuring Covenant Listeners

Listeners lie at the core of Covenant's communication infrastructure, enabling the framework to interact with the compromised hosts, known as "grunts." In this section, we will explore how to configure listeners in Covenant.

Step 1: Listener Configuration

By default, Covenant does not provide any pre-configured listeners. To set up a listener, follow these steps:

1. Click on the "Listeners" option in the left menu to open the Listener page.
2. Click the "Create" button to initiate the creation of a new listener.
3. On the Create HTTP Listener page, update the following values:

- **Name:** Modify the default name to something more readable if desired.
- **BindAddress and ConnectAddress:** Specify an address reachable by the grunts.
- **Use SSL:** If using SSL, provide the SSL certificate file and password.

4. Once the configuration is complete, click the “Create” button to create and enable the listener.

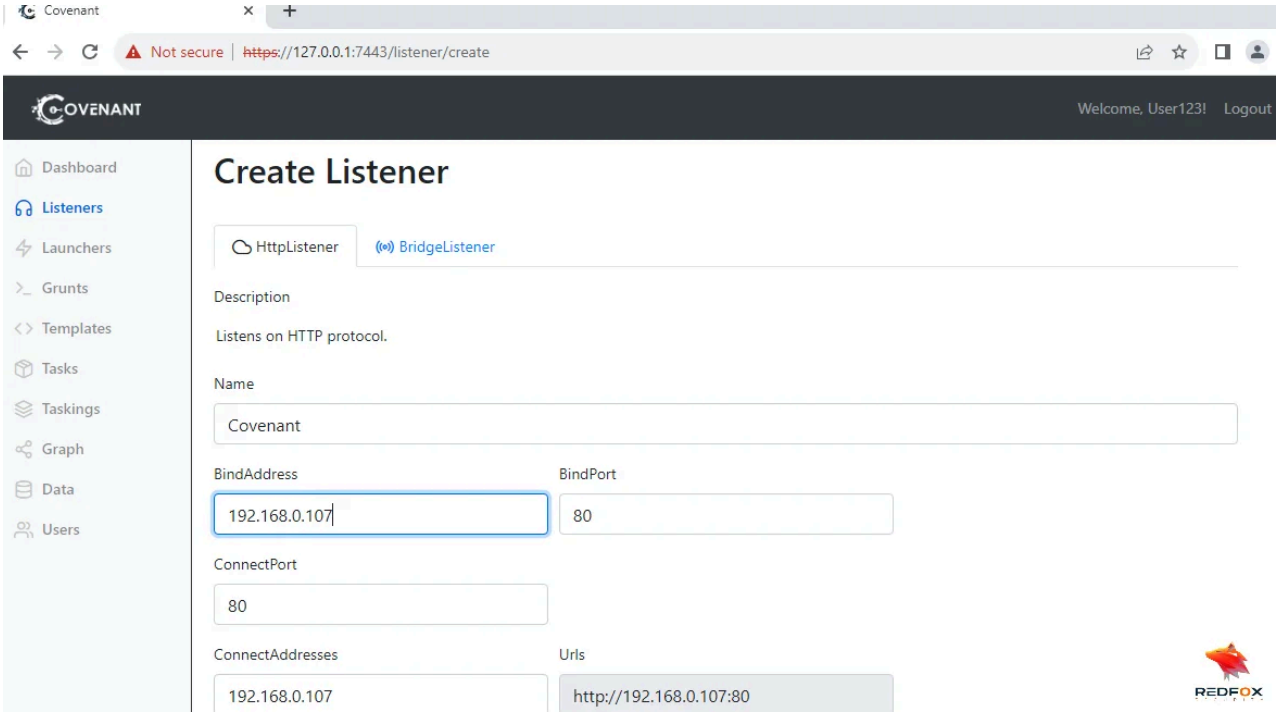

Figure 7: Creating Listener

Section 3: Creating Launchers in Covenant

Launchers serve to transform remote hosts into grunts and establish a connection with the Covenant application. In this section, we will explore the process of creating and executing launchers in Covenant.

Step 1: Choosing a Launcher Type

Covenant offers various launcher types, each catering to different scenarios. For our first launcher, let’s select the Binary launcher, which generates custom binaries. Follow these steps:

1. From the User Management page, click on “Launchers” in the left menu.
2. Select the “Binary” launcher from the list of available types.

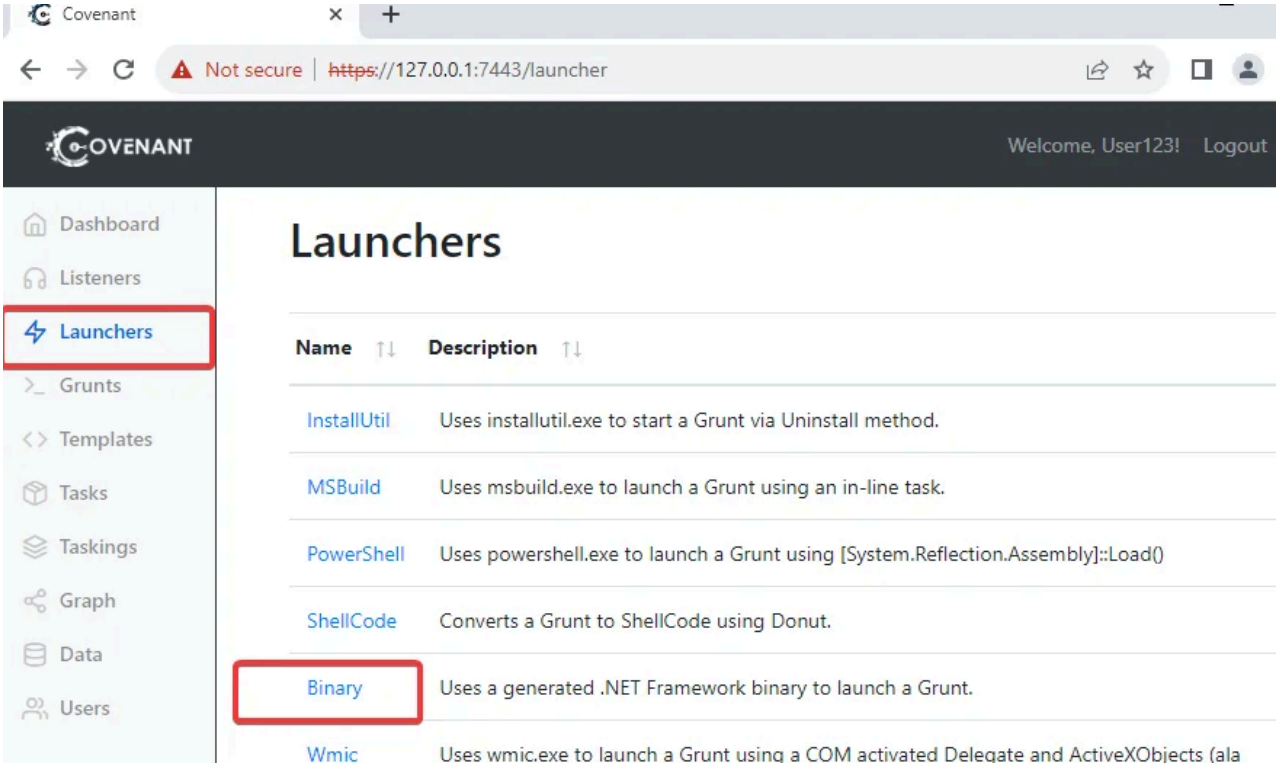

Figure 8: Creating Launcher

Step 2: Launcher Configuration

You will be presented with a configuration page upon selecting the Binary launcher. Configure the launcher by providing the following information:

- **Listener:** Choose the HTTP listener created earlier.
- **Template:** Opt for the pre-configured template “GruntHTTP.”
- **Delay and Jitter:** These settings control the frequency of communication between the grunt and Covenant. The default values are suitable for most scenarios.
- **ConnectAttempts and KillDate:** These settings determine the duration of communication attempts. The default values can be retained.

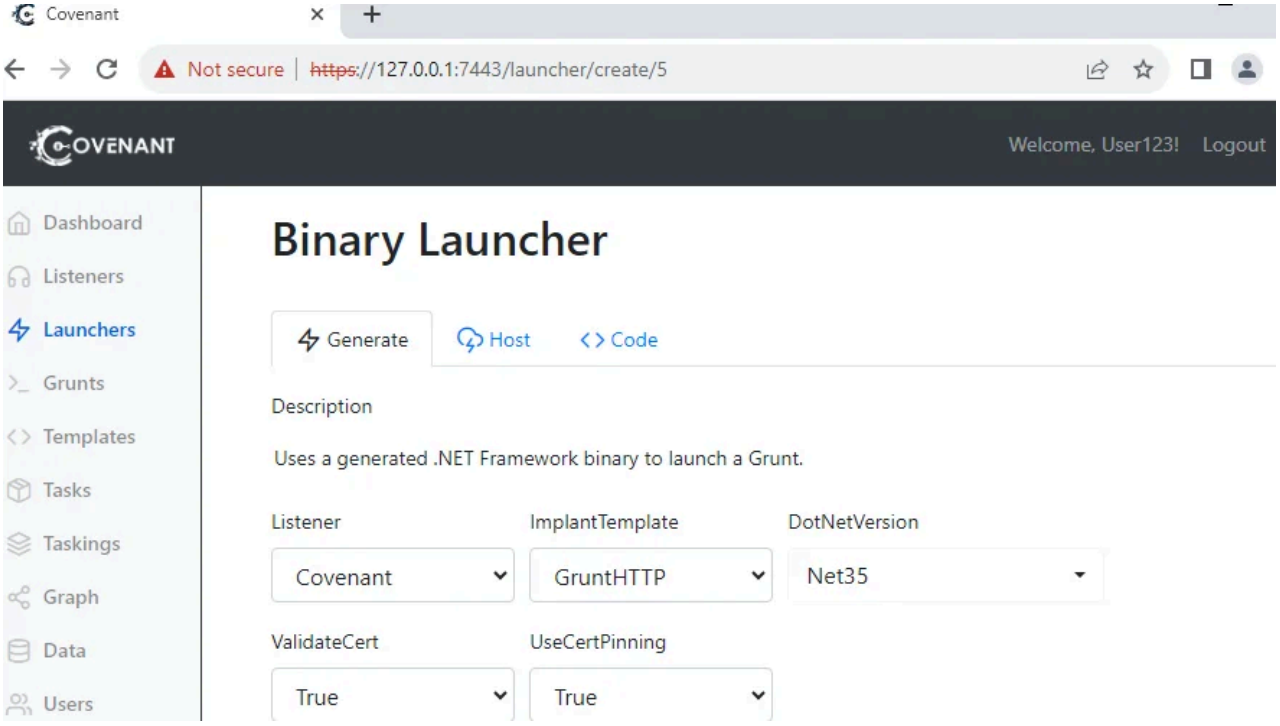

Figure 9: Creating Binary Launcher

Once the configuration is complete, click the “Generate” button to create the launcher. Save the generated launcher payload by clicking “Download” and then “Save File.”

Note: Make sure to disable Windows Defender before downloading the Launcher.

Step 3: Executing the Launcher

Copy the saved launcher file to the desired host and execute it. Once executed, the newly registered grunt will appear on the Covenant dashboard, ready for further assignments.

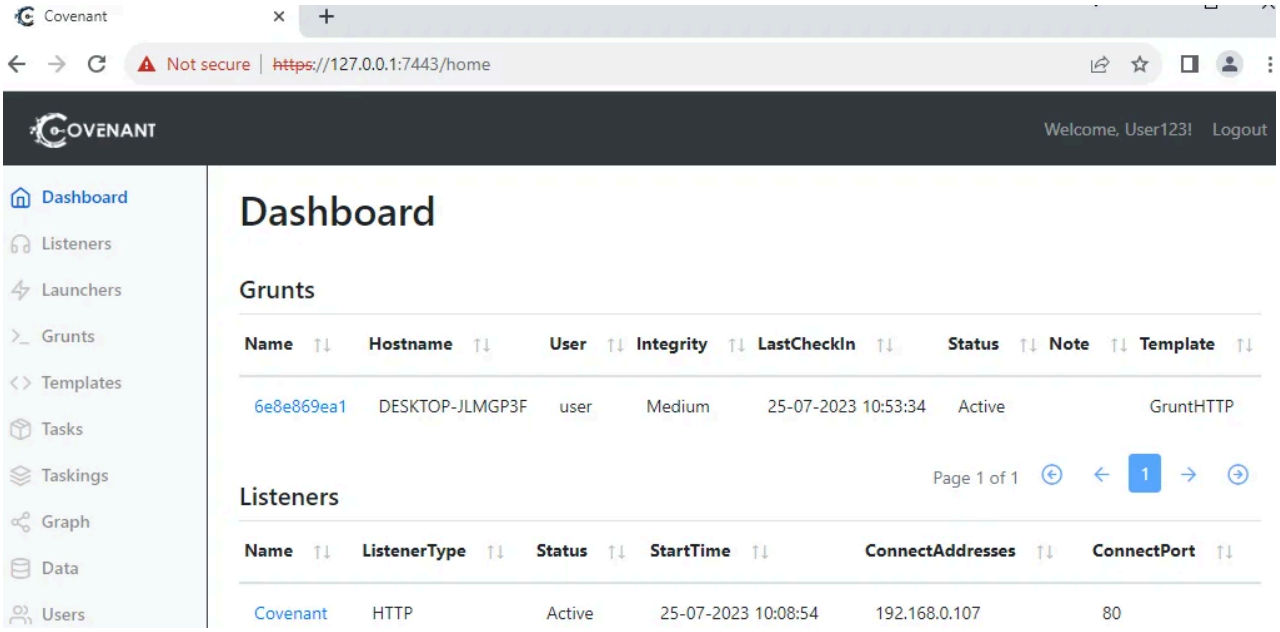

Figure 10: New Grunt

Section 4: Task Assignment in Covenant

With grunts registered and connected to the Covenant application, it's time to assign tasks and leverage the full potential of this C2 framework. In this section, we will explore how to assign tasks to grunts in Covenant.

Step 1: Accessing Grunt Details

From the User Management page, navigate to "Grunts" and open the grunt you wish to assign tasks. The "Info" tab details the grunt's connection status and functionality.

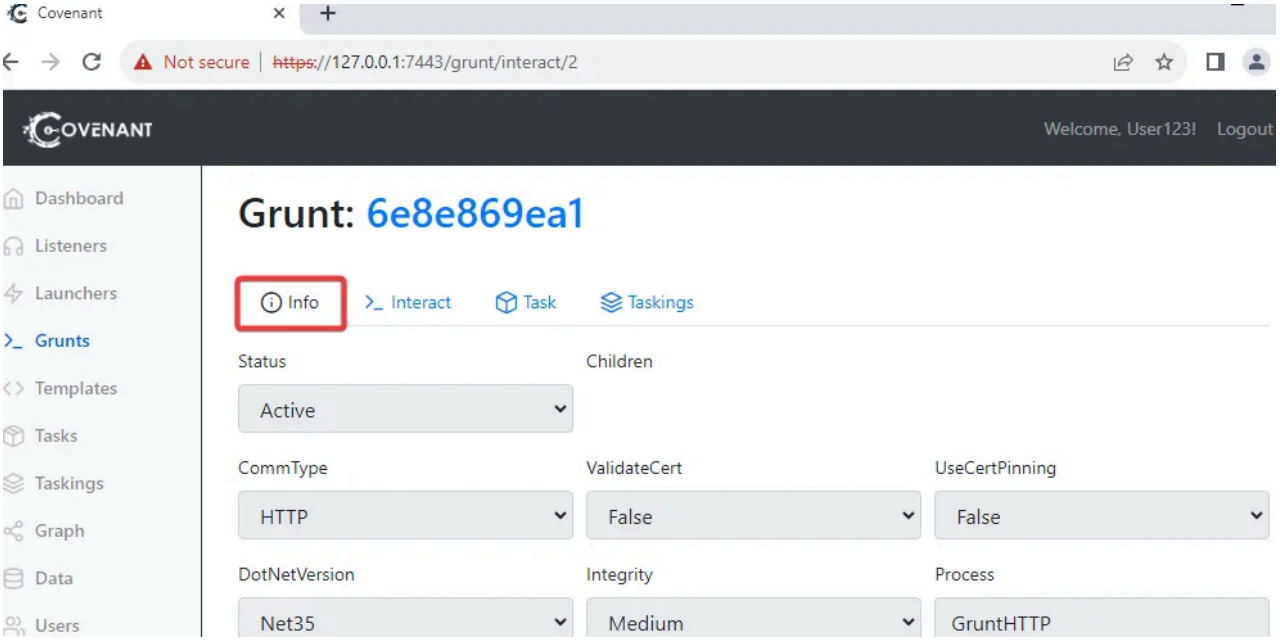

Figure 11: Grunt Info tab

Step 2: Assigning Tasks

To assign a task to the grunt, click the "Task" tab. Covenant offers a wide range of functions, each serving a specific purpose. Let's explore a few examples:

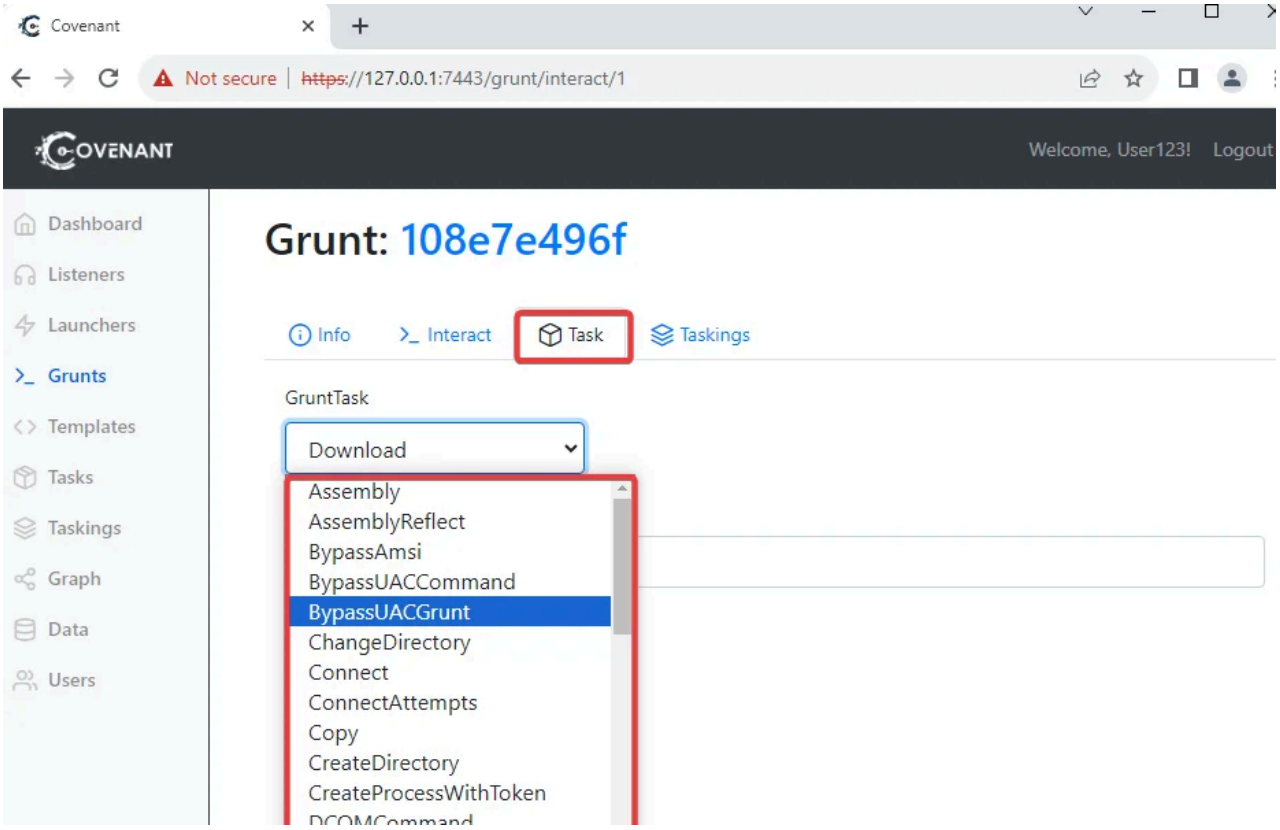

Figure 12: Assigning Task

Example 1: Taking a Screenshot

Suppose you need the grunt to screenshot the active user's desktop. Follow these steps:

1. From the Task dropdown, select "Screenshot."
2. Click the "Task" button to execute the screenshot task on the grunt.

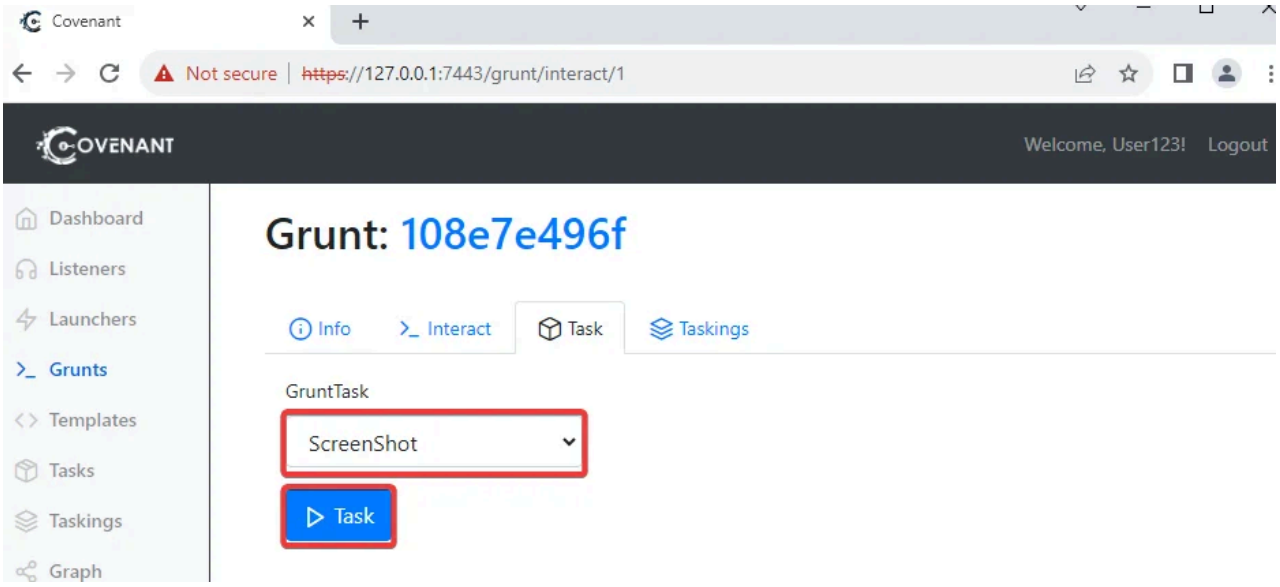

Figure 13: Creating grunt task

Once the task is completed, the updated task status and output will be visible. In this case, the output will be the captured screenshot.

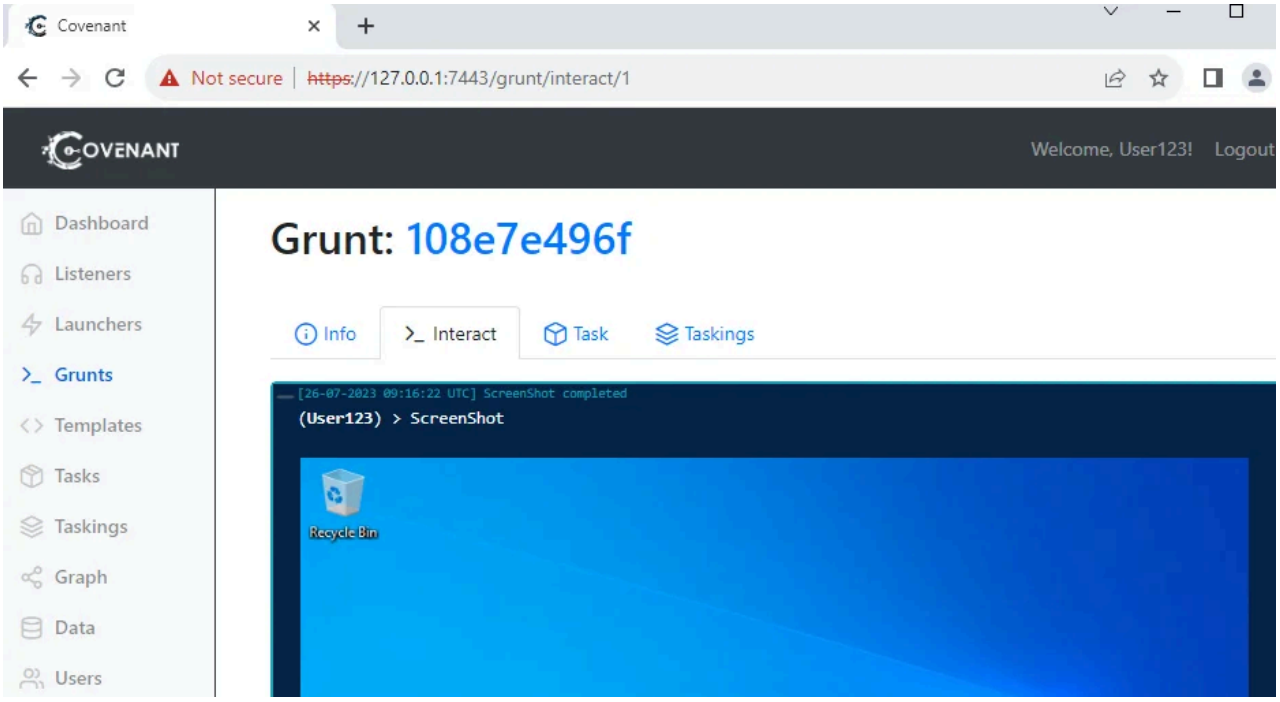

Figure 14: Screenshot

Example 2: Running Mimikatz

Mimikatz is a powerful tool for extracting credentials and performing various post-exploitation activities. Let's assign a Mimikatz task to the grunt:

1. Navigate back to the Tasks tab of the grunt's details page.
2. Select "Mimikatz" from the Task dropdown.

Mimikatz tasks offer customizable input parameters. By default, Covenant pre-populates the "Command" input parameter with the sekurlsa::logonPasswords command, which uncovers passwords stored in memory. Here we are using Mimikatz's "lsadump::sam" command to dump all the sam hashes.

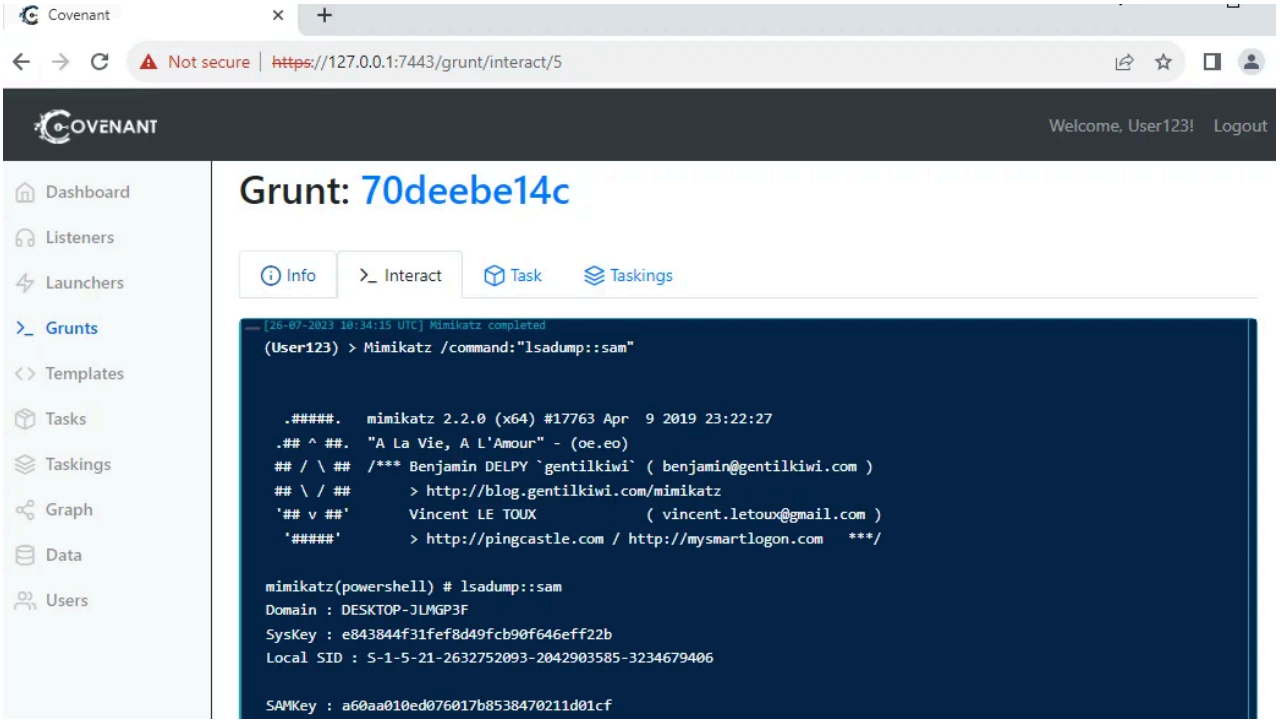

Figure 15: Running Mimikatz

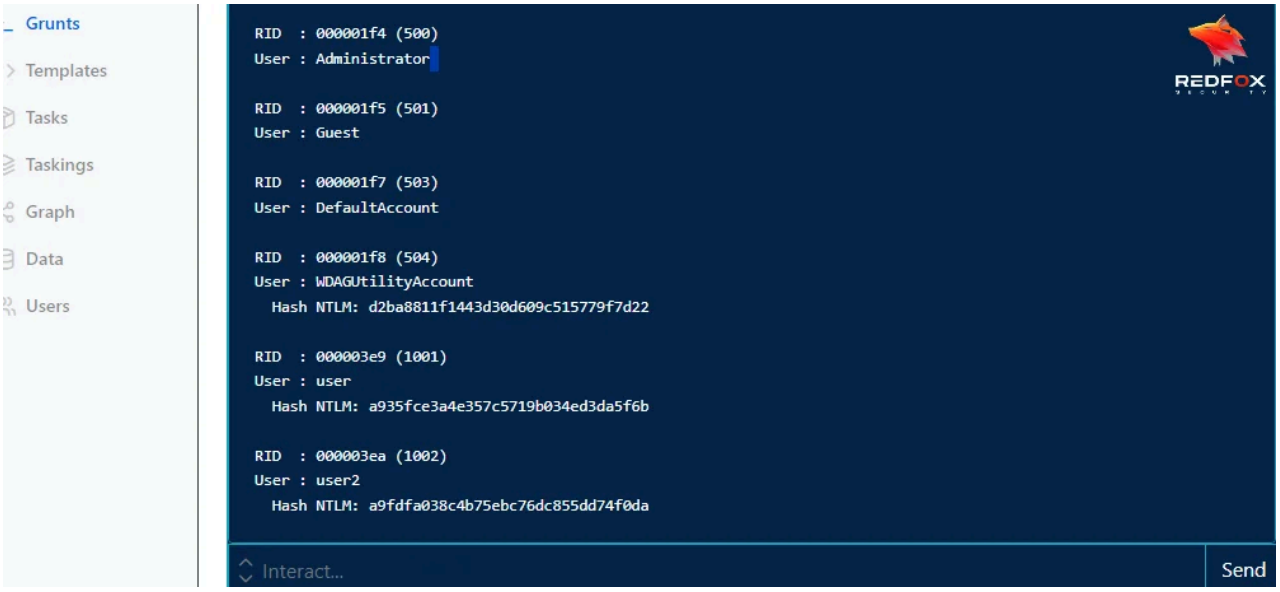

Figure 16: Sam Hashes

Step 3: Task History and Output

Covenant maintains a comprehensive task history, allowing you to review and analyze past executions. Access the Taskings page from the User Management menu to view the history. Clicking on a specific task will provide access to its parameters and full output. Additionally, Covenant saves relevant task output elements on the Data page, including credentials, indicators, downloaded artifacts, and screenshots.

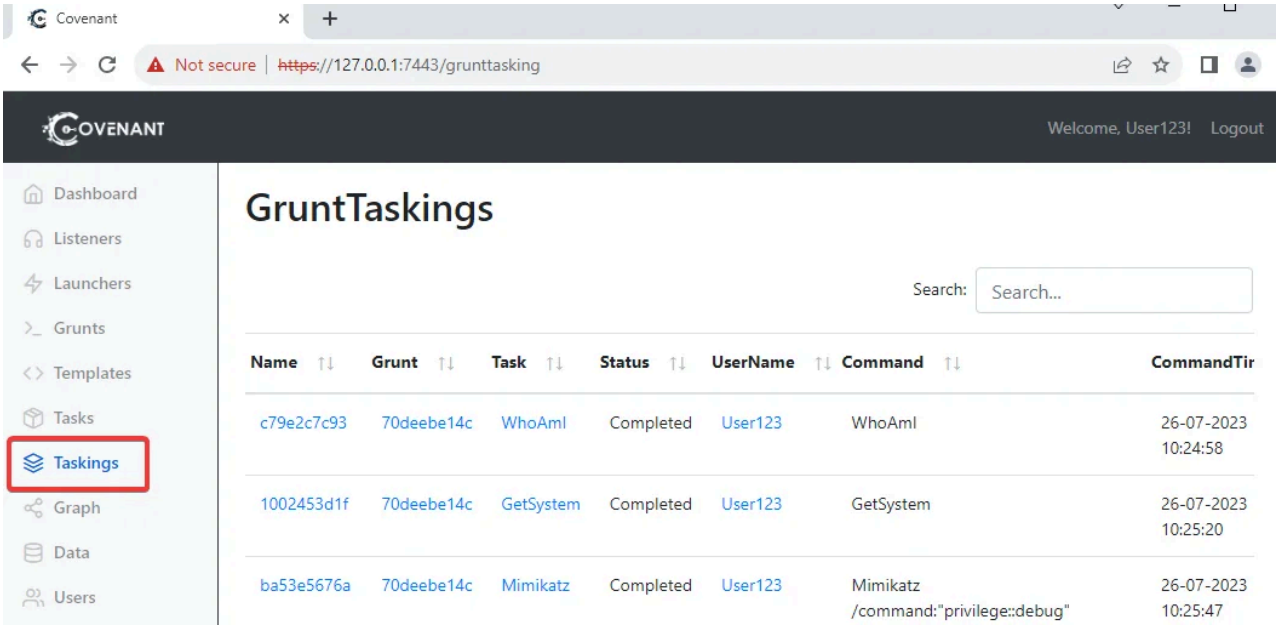

Figure 17: Taskings

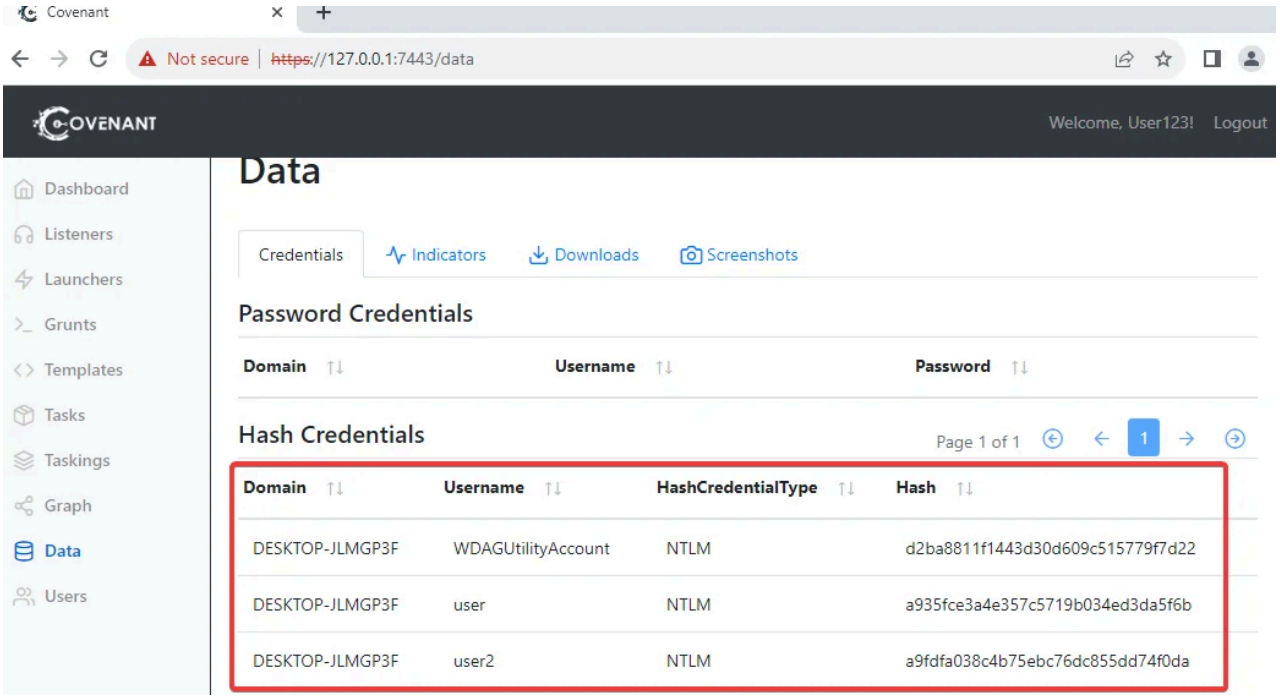

Figure 18: Data

Section 5: Downloading Files with Covenant

Covenant allows downloading files from compromised hosts, providing security professionals with valuable insights and evidence. In this section, we will explore how to download files using Covenant.

Step 1: File Download Task

Suppose you discover a file named MySecretDocument.txt on a compromised host and want to obtain a copy for analysis. Follow these steps:

1. Navigate to the Grunt's Tasks tab.
2. Select the "Download" task from the dropdown menu.

Specify the relevant details, such as the file path and destination. When executed, Covenant will retrieve a copy of the file and make it available on the Downloads tab for further analysis.

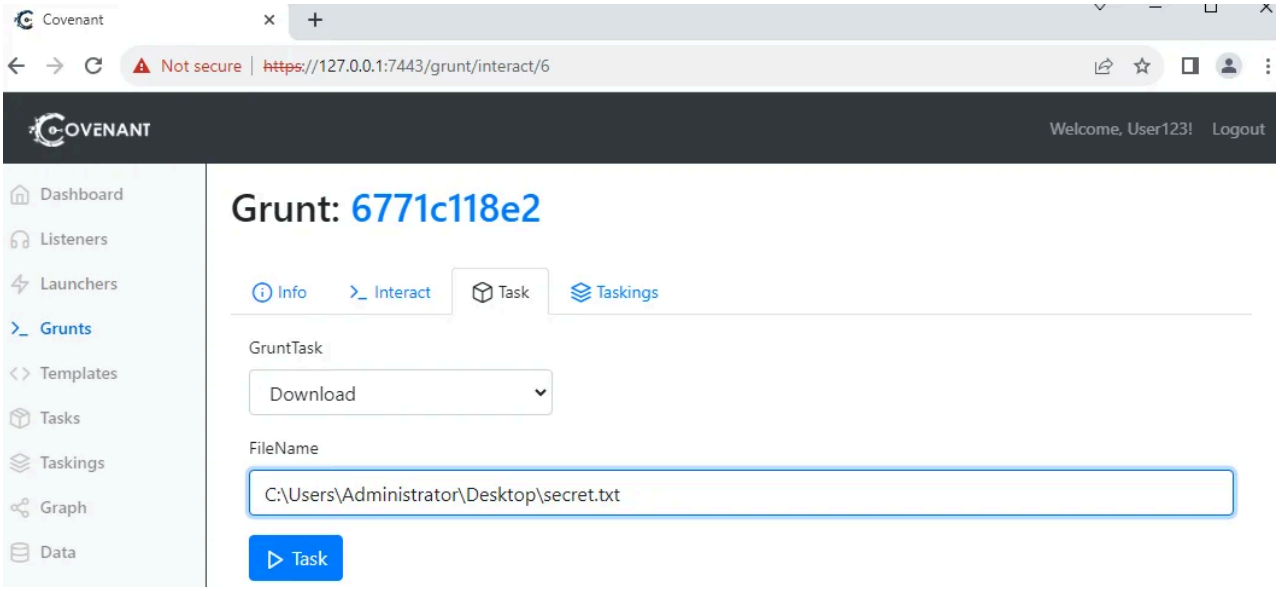

Figure 19: Downloading file

Section 6: Covenant's Potential and Future

Covenant's capabilities extend beyond the examples covered in this guide. With an active open-source community, Covenant continues to evolve, offering an expanding list of tasks and functionalities. As a security professional, leveraging Covenant can significantly enhance your defensive strategies and aid in understanding adversary techniques. By utilizing Covenant's dynamic and adaptable nature, you can stay one step ahead of malicious actors and protect your network effectively.

In conclusion, Covenant is a powerful and versatile C2 framework, offering many post-exploitation possibilities. As you explore its functionalities and delve deeper into its potential, you'll discover the true power of Covenant in bolstering your cybersecurity defenses.

Redfox Security is a diverse network of expert security consultants with a global mindset and a collaborative culture. If you are looking to improve your organization's security posture, **contact us** today to discuss your security testing needs. Our team of security professionals can help you **identify vulnerabilities and weaknesses in your systems, and provide recommendations to remediate them.**

"Join us on our journey of growth and development by signing up for our comprehensive **courses**."

PreviousDOM-Based Cross-Site Scripting
NextThe Importance of Vulnerability Scans and Pen Testing

Recent Blog

September 09, 2025

Is APK Decompilation Legal? What You Need To Know
September 06, 2025
When Hackers Hit the Road: The Jaguar Land Rover Cyberattack
September 05, 2025
This Is the Hacker’s Swiss Army Knife. Have You Heard About It?