

GitHub - cobbr/Covenant: Covenant is a collaborative .NET C2 framework for red teamers.

By cobbr

Archived: 2026-04-05 15:07:23 UTC



Covenant is a .NET command and control framework that aims to highlight the attack surface of .NET, make the use of offensive .NET tradecraft easier, and serve as a collaborative command and control platform for red teamers.

Covenant is an ASP.NET Core, cross-platform application that includes a web-based interface that allows for multi-user collaboration.

The screenshot shows the Covenant web application dashboard. The top navigation bar includes the Covenant logo, the user name 'cobbr', and a 'Logout' link. The left sidebar contains navigation links for Dashboard, Listeners, Launchers, Grunts, Tasks, Taskings, Graph, Data, and Users. The main content area is titled 'Dashboard' and is divided into three sections: Grunts, Listeners, and Taskings.

Grunts

Name	CommType	Hostname	UserName	Status	LastCheckIn	Integrity	OperatingSystem	Process
176a56f1c8	SMB	DESKTOP-F9DQ76G	cobbr	Active	7/18/19 9:21:46 PM	High	Microsoft Windows NT 10.0.17134.0	powershell
31f991ef6c	HTTP	DESKTOP-F9DQ76G	cobbr	Active	7/18/19 9:49:18 PM	High	Microsoft Windows NT 10.0.17134.0	powershell
514c08cc97	SMB	DESKTOP-F9DQ76G	cobbr	Active	7/18/19 9:16:21 PM	High	Microsoft Windows NT 10.0.17134.0	powershell
b564dcaa12	HTTP	DESKTOP-F9DQ76G	cobbr	Active	7/18/19 9:49:15 PM	High	Microsoft Windows NT 10.0.17134.0	powershell

Showing 1 to 4 of 4 entries

Listeners

Name	ListenerType	Status	StartTime	BindAddress	BindPort
62eb6bd841	HTTP	Active	7/18/19 8:57:55 PM	0.0.0.0	80

Taskings

Name	Grunt	Task	Status	UserName	Command	CommandTime	CompletionTime
0903d01960	176a56f1c8	LogonPasswords	Completed	cobbr	LogonPasswords	7/18/19 9:21:11 PM	7/18/19 9:21:21 PM
2c72b6e1ce	31f991ef6c	Connect	Progressed	cobbr	connect localhost gruntsvc	7/18/19 9:08:25 PM	1/1/01 12:00:00 AM
331eedd16c	176a56f1c8	PowerShell	Completed	cobbr	powershell \$PSVersionTable	7/18/19 9:21:26 PM	7/18/19 9:21:30 PM
4f2dc6ff95	514c08cc97	WhoAml	Completed	cobbr	whoami	7/18/19 9:16:07 PM	7/18/19 9:16:10 PM

Quick-Start Guide

Please see the [Installation and Startup](#) guide to get started with Covenant!

The [Wiki](#) documents most of Covenant's core features and how to use them.

Features

Covenant has several key features that make it useful and differentiate it from other command and control frameworks:

- **Intuitive Interface** - Covenant provides an intuitive web application to easily run a collaborative red team operation.
- **Multi-Platform** - Covenant targets .NET Core, which is multi-platform. This allows Covenant to run natively on Linux, MacOS, and Windows platforms. Additionally, Covenant has docker support, allowing it to run within a container on any system that has docker installed.
- **Multi-User** - Covenant supports multi-user collaboration. The ability to collaborate has become crucial for effective red team operations. Many users can interact with the same Covenant server and operate independently or collaboratively.
- **API Driven** - Covenant is driven by an API that enables multi-user collaboration and is easily extendible. Additionally, Covenant includes a Swagger UI that makes development and debugging easier and more convenient.
- **Listener Profiles** - Covenant supports listener “profiles” that control how the network communication between Grunt implants and Covenant listeners look on the wire.

- **Encrypted Key Exchange** - Covenant implements an encrypted key exchange between Grunt implants and Covenant listeners that is largely based on a similar exchange in the Empire project, in addition to optional SSL encryption. This achieves the cryptographic property of forward secrecy between Grunt implants.
- **Dynamic Compilation** - Covenant uses the Roslyn API for dynamic C# compilation. Every time a new Grunt is generated or a new task is assigned, the relevant code is recompiled and obfuscated with ConfuserEx, avoiding totally static payloads. Covenant reuses much of the compilation code from the SharpGen project, which I described in much more detail in a previous post.
- **Inline C# Execution** - Covenant borrows code and ideas from both the SharpGen and SharpShell projects to allow operators to execute C# one-liners on Grunt implants. This allows for similar functionality to that described in the SharpShell post, but allows the one-liners to be executed on remote implants.
- **Tracking Indicators** - Covenant tracks “indicators” throughout an operation, and summarizes them in the Indicators menu. This allows an operator to conduct actions that are tracked throughout an operation and easily summarize those actions to the blue team during or at the end of an assessment for deconfliction and educational purposes. This feature is still in it’s infancy and still has room for improvement.
- **Developed in C#** - Personally, I enjoy developing in C#, which may not be a surprise for anyone that has read my latest blogs or tools. Not everyone might agree that development in C# is ideal, but hopefully everyone agrees that it is nice to have all components of the framework written in the same language. I’ve found it very convenient to write the server, client, and implant all in the same language. This may not be a true “feature”, but hopefully it allows others to contribute to the project fairly easily.

Questions and Discussion

Have questions or want to chat more about Covenant? Join the #Covenant channel in the [BloodHound Gang Slack](#).

Source: <https://github.com/cobbr/Covenant>