


Covenant C2 Fills the Void Left by Empire PowerShell

 blog.netwrix.com/2023/01/27/powershell-empire-covenant

Joe Dibley

Post-exploitation tools are used by threat actors to move laterally inside a network and escalate their privileges in order to steal data, unleash malware, create backdoors and more. Red teams and ethical hackers also use these tools; indeed, simulating the efforts of adversaries plays a key role in implementing effective controls to secure systems, applications and files.

Until recently, one of the most widely used post-exploitation frameworks was PowerShell Empire (PSEmpire). It enabled adversaries to use PowerShell and Python agents to deploy malicious items, ranging from keyloggers to mimikatz, in Windows, Linux and macOS environments while evading detection. However, in 2019, the Empire GitHub Project Page posted that PowerShell Empire was no longer being supported. This was likely due to endpoint detection and response (EDR) tools becoming more effective at detecting malicious PowerShell usage.

Accordingly, many attackers and cybersecurity professionals alike have transitioned to next-generation open source C2 frameworks. This article explores one of these post exploitation tools: Covenant C2.

Handpicked related content:

[Netwrix Webinar | See How Cyberattacks Unfold — and How They Can Be Stopped in Real Time](#)

About Covenant C2

Covenant C2 is a command and control (C&C) framework that makes it easy to exploit web applications and their supporting network environments. This highly scalable, open source framework is available on GitHub. It implements in minutes, even for those with minimal C2 framework experience, and offers an intuitive web-based interface (see Figure 1) that makes it easy to add new agents, data sources and integrations, and to extract data and credentials.

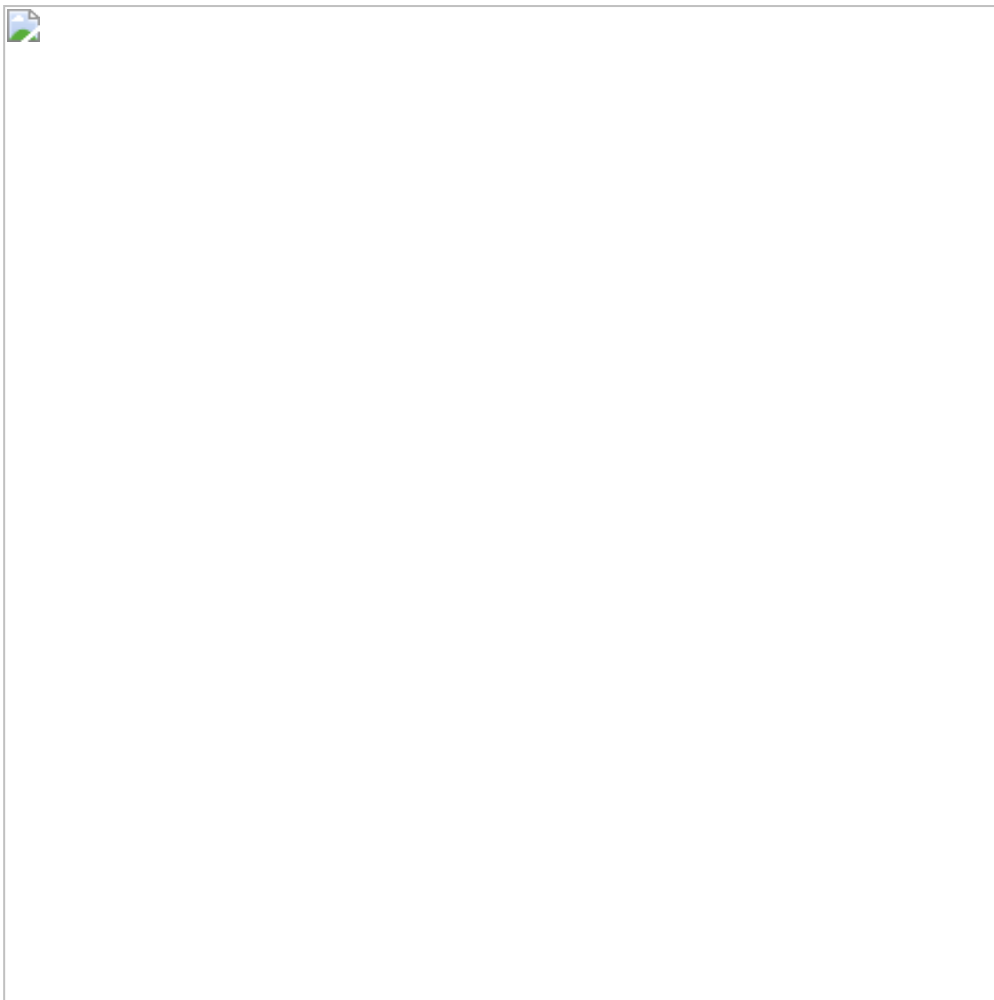


Figure 1. Covenant dashboard (source: [GitHub](#))

Architecture

Covenant is built upon a highly scalable architecture written in C# that enables developers to easily build integrations with the framework. It supports multi-user profiles with authentication, and utilizes lightweight, reliable databases like SQLite and PostgreSQL. Its multi-platform support includes Windows, macOS and Linux distributions such as Kali.

The Covenant Server runs command-and-control functionality, allowing users to operate collaboratively with one another. It utilizes **grunts** — agents that behave much like the grunt soldiers in ancient warfare that were utilized as cannon fodder. These grunts communicate with the Covenant server as they conduct their assigned tasks and avoid detection.

Implementing Covenant

A Covenant server can be up and running in a matter of minutes. On Windows, the only prerequisite is the [.NET Core SDK](#). To build and starts the Covenant ASP .NET Core application, simply run the following commands:

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

```
cd Covenant/Covenant
```

```
dotnet build
```

```
dotnet run
```

Alternatively, you can use a [Docker container](#).

Then browse to the localhost machine using port 7443, and you can start using the Covenant UI using your preferred browser.

Using Covenant

The first task is to set up a listener, as detailed [here](#). Then create a payload to deploy your grunts on remote machines where they will establish communications with the Covenant server.

Then you can use the following pages in the Covenant UI to create, execute and review tasks.

Grunts Page

Use the [Grunts page](#) to review your grunts and have them execute tasks.



Figure 2. Grunt detail page (source: [GitHub](#))

Example: Deploying mimikatz

For example, the video below illustrates how to use a Grunt to deploy mimikatz:

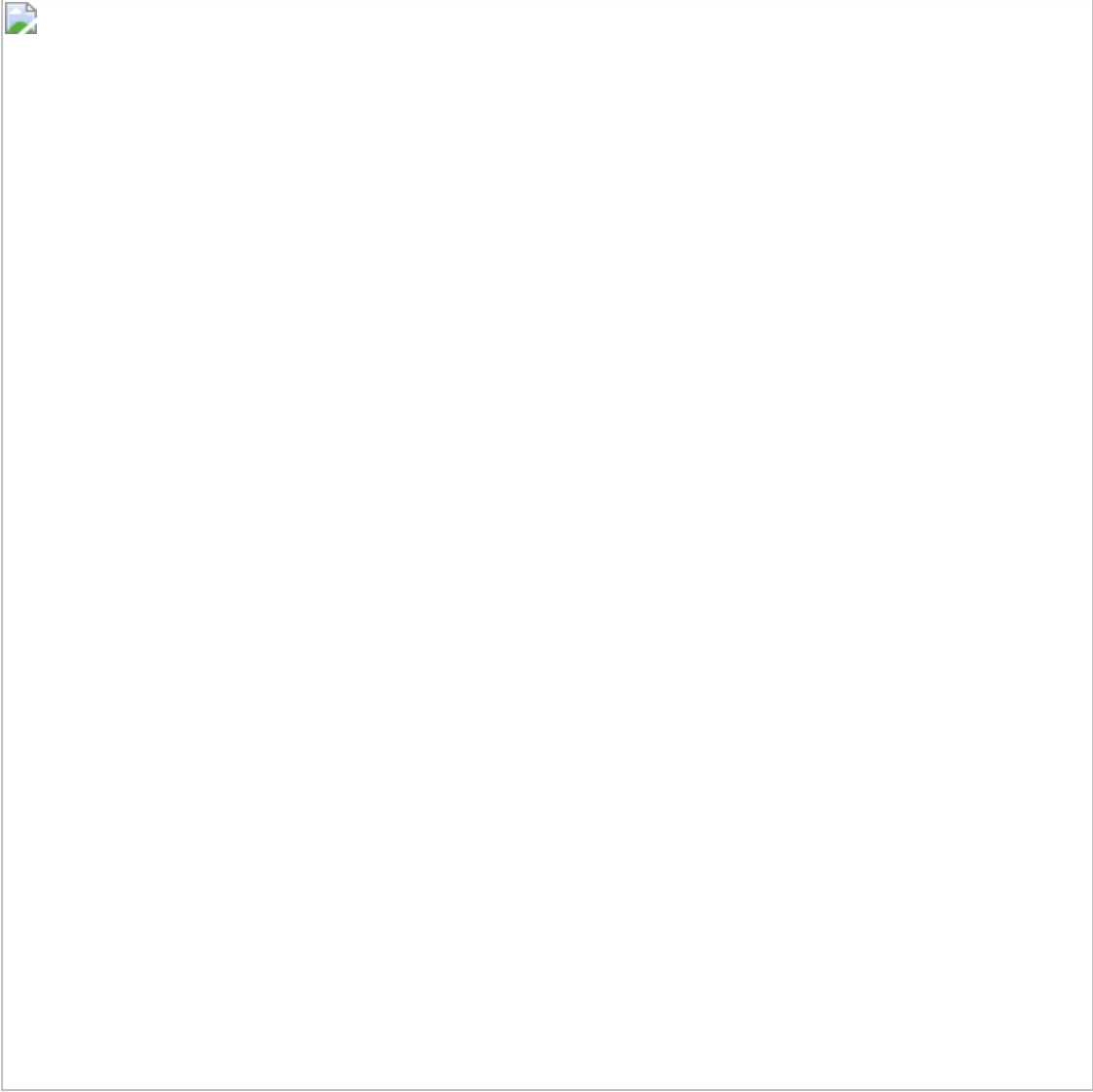


Figure 3. Using a grunt to deploy mimikatz

You can review the output of the executed task using the browser interface.



Figure 4. Reviewing the output of a task performed by a grunt

Tasks Page

The Tasks page provides a sortable and searchable list of all available tasks you can assign to your Grunts.



Figure 5.Tasks page

Taskings Page

The Taskings Page shows the tasks you have assigned to grunts and their status.



Figure 6.Taskings page

Data Page

The purpose of tasks is to capture data, which is summarized on the [Data Page](#). The Indicators tab provides an audit trail of activity, which you can hand off to your blue team to bolster their security efforts.



Figure 7.Data page

Conclusion

Covenant C2 is a great example of how open-source C2 post-exploitation projects have taken the baton from PowerShell Empire to empower red teams. It includes many impressive features that make it ideal for collaborative ethical hacking efforts. There are a lot of great resources on the [Covenant GitHub](#) to help you get started. If the Covenant framework is new to you, it's time to check it out.

Joe Dibley

Security Researcher at Netwrix and member of the Netwrix Security Research Team. Joe is an expert in Active Directory, Windows, and a wide variety of enterprise software platforms and technologies, Joe researches new security risks, complex attack techniques, and associated mitigations and detections.



