Open in app ↗

Sign up   Sign in

Medium   Search   Write

Covenant v0.5

Ryan Cobb · Follow

Published in Posts By SpecterOps Team Members · 5 min read · Jun 4, 2020

✕

Medium

Sign up to discover human stories that deepen your understanding of the world.

Free

✓ Distraction-free reading. No ads.

✓ Organize your knowledge with lists and highlights.

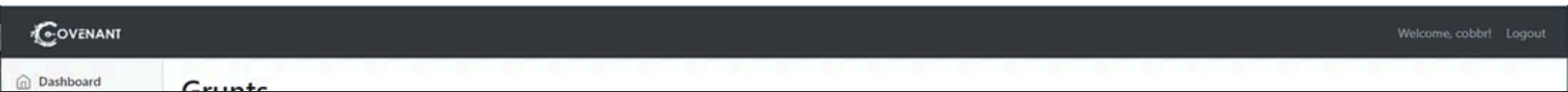✓ Tell your story. Find your audience.

Sign up for free

✦ **Membership**

✓ Read member-only stories

✓ Support writers you read most

✓ Earn money for your writing

✓ Listen to audio narrations

✓ Read offline with the Medium app

Try for 5 $/month

The largest update in v0.5 is migrating the interface over to the Blazor framework. Blazor is a framework for writing client-side web interfaces in C#.

Prior to Blazor, Covenant didn't utilize any real client-side framework like React, Angular, etc. The client-side of the Web UI was written with some one-off JavaScript functions that utilized only JQuery. This made UI components difficult to scale and maintain. Since Covenant is already built on .NET Core, it made a lot of sense to use Microsoft's new client-side framework, Blazor.

been overhauled to take advantage of the real-time features provided by Blazor.

Brutes: The New .NET Core Implant

Covenant has a brand new built-in, cross-platform implant: Brutes! Operators are no longer limited to Grunt implants or to Windows targets.

COVENANT

Welcome, cobbr! Logout

Dashboard   Grunts

Medium

Sign up to discover human stories that deepen your understanding of the world.

Free

✓ Distraction-free reading. No ads.

✓ Organize your knowledge with lists and highlights.

✓ Tell your story. Find your audience.

Sign up for free

✦ **Membership**

✓ Read member-only stories

✓ Support writers you read most

✓ Earn money for your writing

✓ Listen to audio narrations

✓ Read offline with the Medium app

Try for 5 $/month

implant creation more accessible to developers. With the introduction of Brutes, Covenant now has an implant that is written in C#, but **not** based on the .NET Framework at all. The implant is fully supported and built-in to the platform.

As Covenant continues to make steps towards the implant-agnostic model, I'd encourage operators and developers to try their hand at modifying existing implants for their needs, or creating their own personal implant!

It also makes it much easier to contribute built-in Tasks to the project. Before this change, built-in Tasks were described programmatically in Covenant's source code, and required source code changes to add new built-in Tasks. Now, YAML files included under the `Data\Tasks` directory will automatically be parsed and imported when Covenant first starts up. Additional tasks that you would like to use can be imported within the interface.

There are many more updates and fixes in this release than we are able to mention here, but here's a few other minor updates you'll find in v0.5:

- **Author Information** — Author information is included in GruntTasks, so you'll get credit for the Tasks you contribute to Covenant.

- **Hide and Unhide Grunts** — The ability to hide and unhide Grunts has been reworked, so you can easily hide and unhide old Grunts that are no longer in use from the UI.

- **Manual Data Creation** — Credentials and Indicators can be manually

for an upcoming post on .NET Core malware for additional information related to the Brute implant.

Dotnet   Dotnet Core   Blazor   Information Security   Hacking

Medium

Sign up to discover human stories that deepen your understanding of the world.

Free

✓ Distraction-free reading. No ads.

✓ Organize your knowledge with lists and highlights.

✓ Tell your story. Find your audience.

Sign up for free

✦ Membership

✓ Read member-only stories

✓ Support writers you read most

✓ Earn money for your writing

✓ Listen to audio narrations

✓ Read offline with the Medium app

Try for 5 $/month