GitHub - cobbr/SharpSploit: SharpSploit is a .NET post-exploitation library written in C#

By cobbr

Archived: 2026-04-05 20:27:19 UTC

SharpSploit is a .NET post-exploitation library written in C# that aims to highlight the attack surface of .NET and make the use of offensive .NET easier for red teamers.

SharpSploit is named, in part, as a homage to the PowerSploit project, a personal favorite of mine! While SharpSploit does port over some functionality from PowerSploit, my intention is **not** at all to create a direct port of PowerSploit. SharpSploit will be it's own project, albeit with similar goals to PowerSploit.

Intro

You'll find some details and motivations for the SharpSploit project in this introductory blog post.

Documentation

The complete SharpSploit API docfx documentation is available here.

For an easier to read, high-level quick reference and summary of SharpSploit functionality, refer to the SharpSploit - Quick Command Reference.

Credits

I owe a ton of credit to a lot of people. Nearly none of `SharpSploit` is truly original work. `SharpSploit` ports many modules written in PowerShell by others, utilizes techniques discovered by others, and borrows ideas and code from other C# projects as well. With that being said, I'd like to thank the following people for contributing to the project (whether they know they did or not :)):

- Justin Bui (@youslydawg) - For contributing the `SharpSploit.Enumeration.Host.CreateProcessDump()` function.
- Matt Graeber (@mattifestation), Will Schroeder (@harmj0y), and Ruben (@FuzzySec) - For their work on PowerSploit.
- Will Schroeder (@harmj0y) - For the PowerView project.
- Alexander Leary (@0xbadjuju) - For the Tokenvator project.
- James Foreshaw (@tiraniddo) - For his discovery of the token duplication UAC bypass technique documented here.
- Matt Nelson (@enigma0x3) - For his Invoke-TokenDuplication implementation of the token duplication UAC bypass, as well his C# shellcode execution method.
- Benjamin Delpy (@gentilkiwi) - For the Mimikatz project.

- Casey Smith (@subtee) - For his work on a C# PE Loader.
- Chris Ross (@xorrior) - For his implementation of a Mimikatz PE Loader found here.
- Matt Graeber (@mattifestation) - For discovery of the AMSI bypass found here.
- Lee Christensen (@tifkin_) - For the discovery of the PowerShell logging bypass found here.
- All the contributors to www.pinvoke.net - For numerous PInvoke signatures.
