

# C# for PenTest

Joas Antonio

<https://www.linkedin.com/in/joas-antonio-dos-santos>

# C# TOOLS - BOH

- [CasperStager](#)
  - PoC for persisting .NET payloads in Windows Notification Facility (WNF) state names using low-level Windows Kernel API calls.
- [CSEexec](#)
  - An implementation of PSEXEC in C#
- [CSharpCreateThreadExample](#)
  - C# code to run PIC using CreateThread
- [CSharpScripts](#)
  - Collection of C# scripts
- [CSharpSetThreadContext](#)
  - C# Shellcode Runner to execute shellcode via CreateRemoteThread and SetThread Context to evade Get-InjectedThread
- [CSharpWinRM](#)
  - .NET 4.0 WinRM API Command Execution
- [PrintNightmare in CSharp](#)
  - C# and Impacket implementation of PrintNightmare CVE-2021-1675/CVE-2021-34527
- [DnsCache](#)
  - This is a reference example for how to call the Windows API to enumerate cached DNS records in the Windows resolver. Proof of concept or pattern only.
- [Farmer](#)
  - Farmer is a project for collecting NetNTLM hashes in a Windows domain. Farmer achieves this by creating a local WebDAV server that causes the WebDAV Mini Redirector to authenticate from any connecting clients.
- [FreshCookies](#)
  - C# .NET 3.5 tool that keeps proxy auth cookies fresh by maintaining a hidden IE process that nавs to your hosted auto refresh page. Uses WMI event listeners to monitor for InstanceDeletionEvents of the Internet Explorer process, and starts a hidden IE process via COM object if no other IE processes are running.

# C# TOOLS - BOH

- [GoldenTicket](#)
  - This .NET assembly is specifically designed for creating Golden Tickets. It has been built with a custom version of SharpSploit and an old 2.0 alpha (x64) version of Powerkatz.
- [Grouper2](#)
  - Find vulnerabilities in AD Group Policy
- [HTTPS CSharp Server](#)
  - Implementing a Multithreaded HTTP/HTTPS Debugging Proxy Server in C# xref.
- [Inception](#)
  - Provides In-memory compilation and reflective loading of C# apps for AV evasion.
- [InveighZero](#)
  - Windows C# LLMNR/mDNS/NBNS/DNS/DHCPv6 spoofer/man-in-the-middle tool
- [KittyLitter](#)
  - Credential Dumper. It is comprised of two components, KittyLitter.exe and KittyScooper.exe. This will bind across TCP, SMB, and MailSlot channels to communicate credential material to lowest privilege attackers.
- [LittleCorporal](#)
  - LittleCorporal: A C# Automated Maldoc Generator
- [Lockless](#)
  - Lockless allows for the copying of locked files.

# C# TOOLS - BOH

- [MaliciousClickOnceMSBuild](#)
  - Basic C# Project that will take an MSBuild payload and run it with MSBuild via ClickOnce.
- [Minidump](#)
  - The program is designed to dump full memory of the process by specifying process name or process id.
- [MiscTools](#)
  - Miscellaneous Tools
- [NamedPipes](#)
  - A pattern for client/server communication via Named Pipes via C#
- [nopowershell](#)
  - PowerShell rebuilt in C# for Red Teaming purposes
- [OffensiveCSharp](#)
  - Collection of Offensive C# Tooling
- [PurpleSharp](#)
  - PurpleSharp is a C# adversary simulation tool that executes adversary techniques with the purpose of generating attack telemetry in monitored Windows environments.
- [Reg\\_Built](#)
  - C# Userland Registry RunKey persistence

# C# TOOLS - BOH

- RemoteProcessInjection
  - C# remote process injection utility for Cobalt Strike
- Rubeus
  - Rubeus is a C# toolset for raw Kerberos interaction and abuses.
- RunProcessAsTask
- RunasCs
  - RunasCs - Csharp and open version of windows builtin runas.exe
- RunSharp
  - Simple program that allows you to run commands as another user without being prompted for their password. This is useful in cases where you don't always get feedback from a prompt, such as the case with some remote shells.
- SafetyDump
  - SafetyDump is an in-memory process memory dumper.
- SafetyKatz
  - SafetyKatz is a combination of slightly modified version of @gentilkiwi's Mimikatz project and @subTee's .NET PE Loader
- Seatbelt
  - Seatbelt is a C# project that performs a number of security oriented host-survey "safety checks" relevant from both offensive and defensive security perspectives.

# C# TOOLS - BOH

- [self-morphing-csharp-binary](#)
  - C# binary that mutates its own code, encrypts and obfuscates itself on runtime
- [Sharp-InvokeWMIExec](#)
  - A native C# conversion of Kevin Robertsons Invoke-WMIExec powershell script
- [Sharp-Suite](#)
  - fork of FuzzySecurity/Sharp-Suite
- [SharpAdidnsdump](#)
  - c# implementation of Active Directory Integrated DNS dumping (authenticated user)
- [SharpAppLocker](#)
  - C# port of the Get-AppLockerPolicy PS cmdlet
- [SharpAttack](#)
  - SharpAttack is a console for certain things I use often during security assessments. It leverages .NET and the Windows API to perform its work. It contains commands for domain enumeration, code execution, and other fun things.
- [SharpBlock](#)
  - A method of bypassing EDR's active projection DLL's by preventing entry point execution
- [SharpCat](#)
  - C# alternative to the linux "cat" command... Prints file contents to console. For use with Cobalt Strike's Execute-Assembly

# C# TOOLS - BOH

- [SharpClipHistory](#)
  - SharpClipHistory is a .NET application written in C# that can be used to read the contents of a user's clipboard history in Windows 10 starting from the 1809 Build.
- [SharpCloud](#)
  - Simple C# for checking for the existence of credential files related to AWS, Microsoft Azure, and Google Compute.
- [SharpCOM](#)
  - CSHARP DCOM Fun
- [SharpCompile](#)
  - SharpCompile is an aggressor script for Cobalt Strike which allows you to compile and execute C# in realtime. This is a more slick approach than manually compiling an .NET assembly and loading it into Cobalt Strike.
- [SharpCradle](#)
  - SharpCradle is a tool designed to help penetration testers or red teams download and execute .NET binaries into memory.
- [SharpDomainSpray](#)
  - Basic password spraying tool for internal tests and red teaming
- [SharpDoor](#)
  - SharpDoor is alternative RDPWrap written in C# to allowed multiple RDP (Remote Desktop) sessions by patching termsrv.dll file.

# C# TOOLS - BOH

- [SharpDPAPI](#)
  - SharpDPAPI is a C# port of some Mimikatz DPAPI functionality.
- [SharpDump](#)
  - SharpDump is a C# port of PowerSploit's Out-Minidump.ps1 functionality.
- [SharpEdge](#)
  - C# Implementation of Get-VaultCredential
- [SharpHook](#)
  - SharpHook is inspired by the SharpRDPThief project, It uses various API hooks in order to give us the desired credentials.
- [SharpChisel](#)
  - C# Wrapper around Chisel from <https://github.com/jpillora/chisel>
- [SharPersist](#)
  - Windows persistence toolkit written in C#.
- [SharpExcelibur](#)
  - Read Excel Spreadsheets (XLS/XLSX) using Cobalt Strike's Execute-Assembly
- [SharpExec](#)
  - SharpExec is an offensive security C# tool designed to aid with lateral movement. WMIEexec. SMBExec. PSEexec. WMI.
- [SharpFiles](#)
  - C# program that takes in the file output from PowerView's Invoke-ShareFinder and will search through the network shares for files containing terms that you specify.

# C# TOOLS - BOH

- [SharpFinder](#)
  - Searches for files matching specific criteria on readable shares within the domain.
- [SharpFruit](#)
  - A C# penetration testing tool to discover low-hanging web fruit via web requests.
- [SharpGPOAbuse](#)
  - application written in C# that can be used to take advantage of a user's edit rights on a Group Policy Object (GPO) in order to compromise the objects that are controlled by that GPO.
- [SharpHide](#)
  - Tool to create hidden registry keys.
- [SharpInvoke-SMBExec](#)
  - SMBExec C# module
- [SharpLoadImage](#)
  - Hide .Net assembly into png images
- [SharpLocker](#)
  - SharpLocker helps get current user credentials by popping a fake Windows lock screen, all output is sent to Console which works perfect for Cobalt Strike.
- [SharpLoginPrompt](#)
  - This Program creates a login prompt to gather username and password of the current user.
- [SharpLogger](#)
  - Keylogger written in C#

# C# TOOLS - BOH

- [SharpMapExec](#)
  - A sharpen version of CrackMapExec. This tool is made to simplify penetration testing of networks and to create a swiss army knife that is made for running on Windows which is often a requirement during insider threat simulation engagements.
- [SharpNeedle](#)
  - Inject C# code into a running process. Note: SharpNeedle currently only supports 32-bit processes.
- [SharpMove](#)
  - .NET Project for performing Authenticated Remote Execution (WMI, SCM, DCOM, Task Scheduler, Service DLL Hijack, DCOM Server Hijack, Modify Scheduled Task, Modify Service binpath)
- [SharpPack](#)
  - An Insider Threat Toolkit. SharpPack is a toolkit for insider threat assessments that lets you defeat application whitelisting to execute arbitrary DotNet and PowerShell tools.
- [sharppcap](#)
  - Official repository - Fully managed, cross platform (Windows, Mac, Linux) .NET library for capturing packets
- [SharpPrinter](#)
  - Discover Printers
- [SharpRelay](#)
  - Relay hashes over CobaltStrike beacon and impacket ntlmrelayx.py.
- [SharpRoast](#)
  - SharpRoast is a C# port of various PowerView's Kerberoasting functionality.
- [SharpShares](#)
  - Enumerate all network shares in the current domain. Also, can resolve names to IP addresses.
- [SharpSC](#)
  - Simple .NET assembly to interact with services.

# C# TOOLS - BOH

- [SharpSniper](#)
  - Find specific users in active directory via their username and logon IP address
- [SharpSocks](#)
  - Tunnellable HTTP/HTTPS socks4a proxy written in C# and deployable via PowerShell
- [SharpSphere](#)
  - .NET Project for Attacking vCenter
- [SharpSploit](#)
  - SharpSploit is a .NET post-exploitation library written in C# <https://sharpsploit.cobbr.io/api/>
- [SharpSpray](#)
  - SharpSpray a simple code set to perform a password spraying attack against all users of a domain using LDAP and is compatible with Cobalt Strike.
- [SharpSSDP](#)
  - SSDP Service Discovery
- [SharpSword](#)
  - Read the contents of DOCX files using Cobalt Strike's Execute-Assembly
- [SharpTask](#)
  - SharpTask is a simple code set to interact with the Task Scheduler service api and is compatible with Cobalt Strike.
- [SharpView](#)
  - C# implementation of harmj0y's PowerView

# C# TOOLS - BOH

- [SharpWeb](#)
  - .NET 2.0 CLR project to retrieve saved browser credentials from Google Chrome, Mozilla Firefox and Microsoft Internet Explorer/Edge.
- [SharpWMI](#)
  - SharpWMI is a C# implementation of various WMI functionality.
- [SharPyShell](#)
  - SharPyShell - tiny and obfuscated ASP.NET webshell for C# web applications
- [SharpZeroLogon](#)
  - This is an exploit for CVE-2020-1472, a.k.a. Zerologon.
- [SilkETW](#)
  - SilkETW & SilkService are flexible C# wrappers for ETW, they are meant to abstract away the complexities of ETW and give people a simple interface to perform research and introspection. While both projects have obvious defensive (and offensive) applications they should primarily be considered as research tools.
- [SneakyService](#)
  - A simple, minimal C# windows service implementation that can be used to demonstrate privilege escalation from misconfigured windows services.

# C# TOOLS - BOH

- [SpaceRunner](#)
  - This tool enables the compilation of a C# program that will execute arbitrary PowerShell code, without launching PowerShell processes through the use of runspace.
- [Stracciatella](#)
  - OpSec-safe Powershell runspace from within C# (aka SharpPick) with AMSI and Script Block Logging disabled at startup
- [taskkill](#)
  - This is a reference example for how to call the Windows API to enumerate and kill a process similar to taskkill.exe. This is based on (incomplete) MSDN example code. Proof of concept or pattern only.
- [TCPRelayInjector2](#)
  - Tool for injecting a "TCP Relay" managed assembly into an unmanaged process.
- [TikiTorch](#)
  - Process Injection. The basic concept of CACTUSTORCH is that it spawns a new process, allocates a region of memory, then uses CreateRemoteThread to run the desired shellcode within that target process. Both the process and shellcode are specified by the user.
- [TrustJack](#)
  - Yet another PoC for <https://www.wietzebeukema.nl/blog/hijacking-dlls-in-windows>.
- [Watson](#)
  - Enumerate missing KBs and suggest exploits for useful Privilege Escalation vulnerabilities

# .NET Security

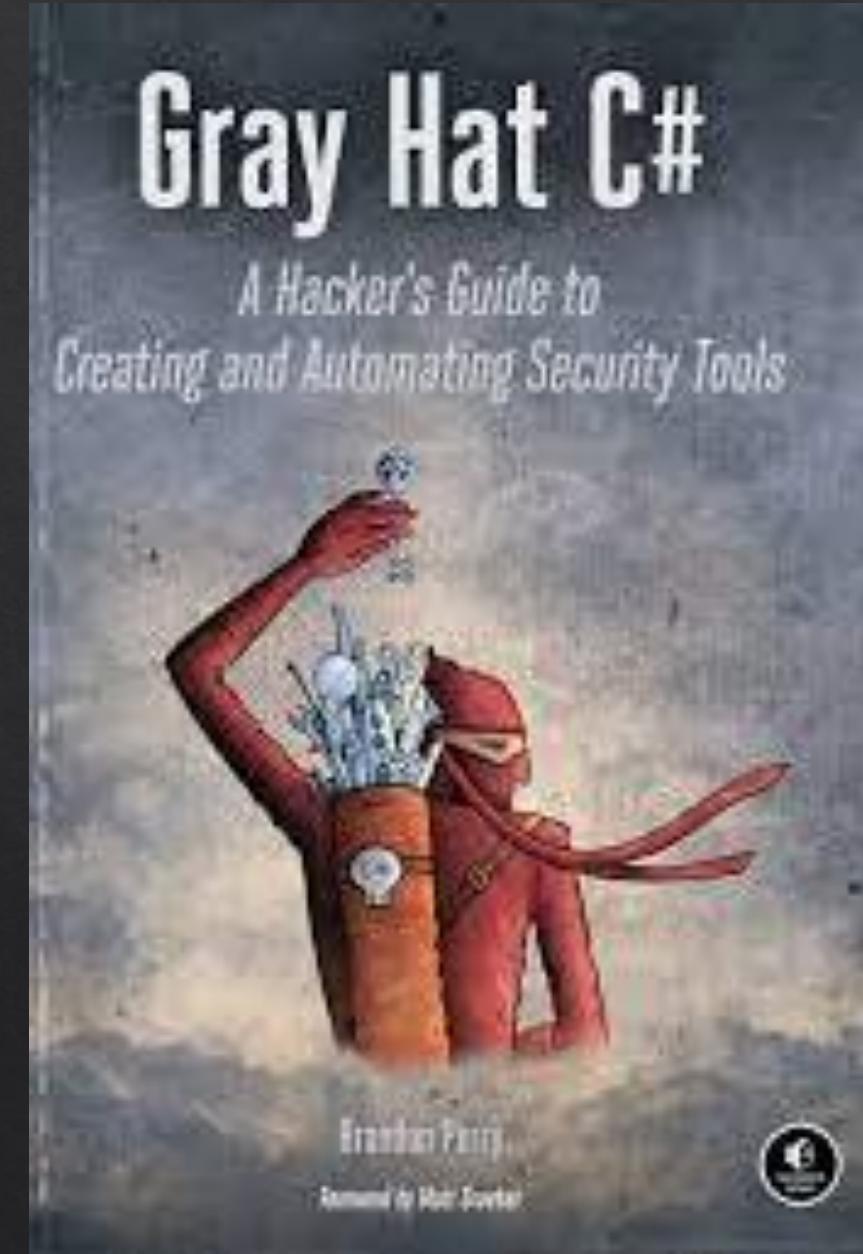
- <https://github.com/guardrailsio/awesome-dotnet-security>
- [https://cheatsheetseries.owasp.org/cheatsheets/DotNet Security Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/DotNet_Security_Cheat_Sheet.html)
- <https://visualstudiomagazine.com/articles/2019/10/22/top-net-attacks.aspx>
- <https://www.toptal.com/security/10-most-common-web-security-vulnerabilities>
- <https://beyondsecurity.com/scan-pentest-network-vulnerabilities-net-framework-allows-code-execution-ms11-028.html>
- <https://www.youtube.com/watch?v=anCRjM7meSo>
- <https://www.youtube.com/watch?v=HMEodGGpzo0>
- <https://resources.infosecinstitute.com/topic/net-penetration-testing-test-case-cheat-sheet/>
- <https://www.youtube.com/watch?v=fCfneKhiYBw>
- <https://www.youtube.com/watch?v=CjoAYsITKX0>
- <https://www.youtube.com/watch?v=ZqARpaB39TY>
- <https://www.youtube.com/watch?v=lb3jnD158NI>
- <https://www.youtube.com/watch?v=M8VeTHB08ao>

# Gray Hat C#

[https://t.me/library\\_Sec/596](https://t.me/library_Sec/596)

[https://github.com/brandonprry/gray\\_hat\\_csharp\\_code](https://github.com/brandonprry/gray_hat_csharp_code)

<https://github.com/tbhaxor/csharp-and-infosec>



# C2 – Command and Control

- <https://github.com/SharpC2/SharpC2>
- <https://github.com/cobbr/Covenant>
- <https://github.com/tcostam/awesome-command-control>
- <https://github.com/EnginDemirbilek/NorthStarC2>
- <https://github.com/AdvancedHacker101>
- <https://github.com/madSimonJ/CSharpHacks>
- <https://www.zeropointsecurity.co.uk/c2-dev-csharp/overview>
- <https://shogunlab.gitbook.io/building-c2-implants-in-cpp-a-primer/>

T.me/Library\_Sec

# C# Fundamentals

- <https://github.com/pwittchen/learning-csharp>
- <https://github.com/dotnet/training-tutorials>
- <https://github.com/chrisvasqm/csharp-beginners>
- <https://github.com/myilm/.Net-Coding-Fundamentals>
- <https://www.youtube.com/watch?v=GhQdIIFylQ8>
- <https://www.youtube.com/watch?v=gfkTfcpWqAY>
- <https://www.youtube.com/watch?v=pSiLHe2uZ2w>
- <https://www.youtube.com/watch?v=wLg-XdAmrak>