Secure Systems Design and Technology Development. Veliko Tarnovo, Bulgaria
Proceedings of the International Scientific Conference, 2026. Volume III, 215-218

Why Did Threat Actors Replace PoshC2 with SliverC2? Comparison of Two Great C2 Frameworks

Dimitar Nikolov
IT Department
Nikola Vaptsarov Naval Academy
Varna, Bulgaria
d.nikolov@naval-acad.bg

***Abstract* – This paper presents a comparative analysis of the command-and-control (C2) frameworks PoshC2 and SliverC2, examined within the context of evolving adversarial tradecraft, shifting infrastructure paradigms and the increasing maturity of defensive capabilities. While C2 frameworks are traditionally associated with the post-exploitation phase, this study adopts a broader perspective, arguing that framework selection reflects not only tactical requirements but also strategic and operational adaptations to a rapidly changing cyber environment. The analysis is based on the recent transformations in enterprise infrastructure, notably the transition from predominantly on-premises, Active Directory-centric Windows environments to complex, distributed, and hybrid cloud ecosystems. This evolution, combined with the expanded attack surfaces across critical sectors, has imposed new constraints on adversary operations. In parallel, defensive practices have become increasingly proactive and intelligence-driven, with widespread adoption of EDR, XDR and SIEM solutions. These developments have contributed to a significant reduction in attacker dwell time and increased detection of well-established tools, techniques, and procedures (TTPs). Within this context, the paper contrasts the architectural and operational characteristics of PoshC2 and SliverC2. PoshC2 is a mature, PowerShell-centric framework optimized for Windows environments, offering extensive post-exploitation functionality but exhibiting increased detectability due to its reliance on heavily monitored technologies. In contrast, SliverC2 represents a modern, Golang-based, cross-platform approach, emphasizing dynamic payload generation, modular extensibility, and advanced communication protocols. The findings of this paper indicate that the shift from PoshC2 to SliverC2 reflects adaptive adversary behavior driven by environmental complexity and defensive pressure, rather than a simple tool substitution. Contemporary offensive operations are inherently multi-framework, with tooling selected dynamically based on operational requirements and target characteristics.**

Keywords – Adversarial tradecraft, command-and-control frameworks, cyber operations, infrastructure paradigms, post-exploitation.

I. INTRODUCTION

When talking about command and control (C2) frameworks they are often viewed in the context of post exploitation. While that is correct from attacker's tactical point of view, since C2 frameworks are used primarily at this stage, in order to fully understand why hackers shifted form using PoshC2 to SliverC2, things like strategic priorities shifts, as well as attackers and defenders changes in tradecraft and TTP's should be taken in consideration. This article intends to analyze the approaches and capabilities offered by both frameworks and makes a comparison of the advantages and disadvantages they have.

Some of the major changes that happened to infrastructure in the recent years is the shift from totally on premises, mainly Active Directory based, Windows environments to a highly technologically and geographically diversified cloud based and hybrid environments. Complexity of the infrastructure has also dramatically increased. Mobile devices, BYOD, laptops and tablets are much more common including in critical infrastructure environments, in the energy, financial and transportation sectors. APT's like Salt Typhoon have been specifically targeting telecommunication companies, in essence positioning themselves for service supply chain attacks against critical infrastructure.

On the other hand, defensive tradecraft has evolved and is currently much more threat hunt and cyber threat intelligence oriented and driven. PowerShell, WMI, CIM and commonly used by C2 network protocols like HTTP, HTTPS, DNS and SMB use are heavily monitored. This combines with the enhanced use of EDR, XDR and SIEM by the SOC teams, creating dynamic and highly

Online ISSN 3044-7771
https://doi.org/10.68302/std2026.vol3.212

environment for the hackers. Dwell times in the recent years have gone down from over 300 days to 10 leaving much shorter window of opportunity for the adversary.

II. POSHC2

PoshC2 was developed by Nettitude (now LRQA) and first released in July 2016 [1]. It is a multi-user, proxy-aware, command-line C2 framework focused on Windows exploitation and post-exploitation, with PowerShell, C#, and Python capabilities. The server is written in Python 3, agents are PowerShell- and C#-based. Version 8 introduced a Linux agent and Beacon Object File (BOF) execution via Joel Snape's RunOF in 2022. In-memory execution of PowerShell and C#/.NET assemblies is provided through a module system invoked by .NET reflection, though this requires additional configuration. Designed for quiet operation using fileless, in-memory implants and encrypted HTTP/S communication, PoshC2 ships pre-configured modules for Mimikatz, keylogging, lateral movement via WMI/PsExec, and automated Active Directory enumeration.

These capabilities, together with out-of-the-box download cradles and shellcode of multiple variants, made PoshC2 attractive to operators when it appeared, and it became a popular choice alongside other heavy PowerShell users such as Empire. Over the following years, however, Microsoft systematically eroded the advantages that PowerShell-centric tooling depended on, introducing in particular:

1) System-wide transcription - a Group Policy–enabled auditing mechanism, introduced in Windows PowerShell 5, that records commands, script blocks, and output to text files for every user on a host.

2) Script-block logging - a runtime mechanism that captures the full content of PowerShell script blocks as they are processed by the engine, yielding far richer records than command-line logging.

3) The Antimalware Scan Interface (AMSI) - introduced in 2015, which gives antimalware products visibility into the PowerShell engine, script hosts (wscript.exe, cscript.exe), Office macros, the .NET Framework, and WMI components frequently abused in living-off-the-land and fileless techniques [3].

4) Constrained Language Mode (CLM) - which restricts advanced PowerShell features-custom .NET methods, COM objects, and Windows API calls-typically enforced through AppLocker or Windows Defender Application Control (WDAC) [4].

Individually, each of these controls is bypassable, taken together they make the Windows environment markedly more hostile to PowerShell abuse. The pre-built download cradles, shellcode, and droppers that were once a PoshC2 advantage became largely impractical without heavy customisation and obfuscation. This shift-combined with the large number of public signatures for PoshC2 implants, intensive PowerShell monitoring and restriction, improved antimalware engines such as Microsoft Defender, and routine hunting for WMI/PsExec lateral-movement indicators-created pressure for a different approach. Operators moved toward flexible in-process use of assemblies and avoidance of fork-and-run execution (an approach popularised by Cobalt Strike's BOFs), and toward implants able to run on Linux, Unix, and macOS as well as in cloud and hybrid environments. A minor but real operator-experience drawback also worked against PoshC2: interacting with an agent and viewing its output typically requires two separate console windows or tabs.

III. SLIVERC2

There is a recurring irony in this space: several of the most capable C2 frameworks were created and are maintained by security companies-Nettitude/LRQA, Bishop Fox, Fortra-and, although intended for adversary emulation, were well-crafted enough to be adopted by real threat actors. SliverC2 is a case in point. It was created by Bishop Fox and introduced in June 2019 as an open-source, Go-based C2 framework for red-team operations and penetration testing, gaining traction from around 2020 for modern features such as mutual TLS (mTLS) and dynamic code generation. It is actively maintained and released under the GPLv3 licence [2].

Sliver is modular and cross-platform, provides native binaries, and supports multiple C2 protocols including HTTP, HTTPS, DNS, and WireGuard. It generates a unique, dynamically compiled implant for each session [5], which complicates signature-based detection. Go has become popular in offensive security and is not yet as heavily instrumented at the host level as PowerShell. Sliver's 'Armory' package system supports extension, and several built-in enumeration commands deliberately avoid classic, well-signatured utilities such as whoami and systeminfo, instead using BOFs and Windows API calls to obtain equivalent information while reducing detection. Sliver also offers several mechanisms to limit or avoid the fork-and-run pattern that seasoned defenders watch for.

IV. COMPARATIVE ANALYSIS

To make the comparison rigorous rather than impressionistic, Table I contrasts the two frameworks across ten fixed dimensions. The discussion that follows interprets the differences that bear most directly on operator selection under present-day defensive pressure.

The most consequential difference is the runtime. PoshC2's reliance on PowerShell and .NET places its core activity on the most heavily instrumented execution surfaces of modern Windows, where transcription, script-block logging, AMSI, and CLM converge. Sliver's Go runtime sits on a less mature host-instrumentation surface and compiles to native binaries for Windows, Linux, and macOS, aligning with the cross-platform, cloud-and-hybrid [12]. For operations that must traverse non-Windows hosts, this is a decisive advantage, for Windows-only, Active-Directory-rich targets, PoshC2's mature AD tradecraft remains relevant.

Dimension	PoshC2	SliverC2
Developer & first release	Nettitude (now LRQA); released July 2016.	Bishop Fox; released June 2019.

Dimension	PoshC2	SliverC2
Implementation language / runtime	Server in Python 3; agents in PowerShell and C#/.NET.	Server and implants in Go (Golang), with cross-compiled native binaries.
Primary OS focus	Windows-centric; a Linux agent and BOF execution were added in v8 (2022).	Cross-platform by design: Windows, Linux, and macOS first-class targets.
Implant generation	Pre-built download cradles, shellcode, and droppers of several variants out of the box.	Per-session, dynamically compiled implants, frustrating static signatures.
C2 protocols	HTTP/HTTPS (encrypted), with proxy awareness.	HTTP, HTTPS, DNS, mTLS, and WireGuard.
Post-exploitation modules	Built-in Mimikatz, keylogging, AD enumeration, lateral movement via WMI/PsExec.	Modular 'Armory' package system; enumeration via BOFs and Windows API calls that avoid signatured commands.
Detection surface (primary)	PowerShell, WMI/CIM, .NET reflection: technologies heavily instrumented on modern Windows.	Go runtime artefacts and large static binaries; less mature host-level instrumentation than PowerShell.
Fork-and-run posture	Module system relies on .NET reflection; reducing fork-and-run requires manual tuning.	Provides several mechanisms to limit or avoid fork-and-run execution.
Operator interface	CLI; interacting with an agent and viewing its output typically needs two windows/tabs.	Single console with both interactive and scripting (operator) modes.
License / maintenance	Open source; maintained by LRQA.	Open source (GPLv3); actively maintained by Bishop Fox.

PoshC2's strength of shipping ready-made cradles, shellcode, and droppers became a liability as those artefacts accumulated public signatures. Sliver's per-session dynamic compilation produces a distinct implant each time, which raises the cost of static, signature-based detection. This advantage is not unconditional: in its default configuration Sliver still leaves substantial host- and network-level footprints, and prevalence telemetry shows that default deployments remain detectable [6].

Both frameworks provide credential access, enumeration, and lateral movement. The differentiator is how that activity manifests to defenders. PoshC2's modules lean on .NET reflection and, where lateral movement uses WMI/PsExec, generate well-understood indicators. Sliver's use of BOFs and direct Windows API calls for enumeration, together with mechanisms to limit fork-and-run, is designed precisely to deny defenders those high-signal behaviours. This mirrors the broader migration toward in-process execution that Cobalt Strike's BOF model popularised.

Kaspersky's quarterly vulnerability-and-exploit reports publish rankings of the C2 frameworks most frequently observed in APT activity. Across 2025 these rankings consistently place Sliver in the top tier-among the most-used frameworks in the first half of the year, and holding the top position in the Q4 2025 analysis, ahead of Mythic and Havoc, [10], [11]. PoshC2 does not appear in these top-ranked lists. Independent vendor telemetry tells a complementary story: Red Canary identified Sliver, Mythic, and Brute Ratel as frameworks to watch as adversaries sought alternatives to more heavily detected tooling [13]. Taken together, the prevalence data substantiate Sliver's ascent to the front rank of observed C2 tooling while PoshC2 remains, at most, a secondary presence.

Beyond aggregate counts, specific intrusions document movement toward Sliver as established tooling became more detectable. Reporting links APT29 (associated with Russia's SVR) to Sliver for maintaining access following the SolarWinds intrusion, as noted in a 2021 NCSC advisory, a prolific ransomware affiliate tracked by Microsoft is reported to have migrated to Sliver as Cobalt Strike detection improved and the financially motivated actor TA551/Shathak distributed Sliver via macro-laden phishing [9]. The DFIR Report further documented Sliver deployed as a hands-on-keyboard payload ahead of ransomware in a 2024 intrusion [7]. The common pattern in these cases is migration away from tooling whose indicators defenders had learned, toward Sliver's lower-profile, cross-platform implants.

The strongest caution against an over-stated 'replacement' narrative comes from incident-response data. In August 2024 The DFIR Report analysed adversary infrastructure-active intermittently since at least September 2023-on which the operators used PoshC2 and Sliver together as their primary C2 toolkits, alongside Ngrok and SystemBC, in activity assessed with medium confidence as ransomware-related. This is direct evidence that PoshC2 was not discarded but retained as one component of a multi-framework toolkit. The honest reading of the combined evidence is therefore a reweighting-Sliver rising [8] to the front rank while PoshC2 persists in a situational, secondary role-rather than a clean one-for-one substitution.

C2-prevalence telemetry is subject to detection and sampling bias. Frameworks that are easier to detect, or that are run in default configurations, are over-represented relative to well-obfuscated or privately forked implants, conversely, heavily customised Sliver builds reported by incident responders may evade the very counts that rank Sliver highly. Vendor visibility also differs: telemetry weighted toward particular customer bases can rank Cobalt Strike or Metasploit above Sliver [14]. Several migration cases rest on secondary CTI reporting rather than primary disclosure. These rankings should therefore be read as directional rather than absolute, and the conclusions below are framed accordingly.

V. CONCLUSION

Is PoshC2 gone? No, it is still a great C2 framework by all means. Truth is modern offensive cyber operations

involve usually more than one tool and C2 frameworks are not an exemption. After gaining initial foothold, a smaller, dispensable, or more applicable frameworks and implants are used like Nimplant or PoshC2 and once internal enumeration, recon, privilege escalation and some sort of persistence is achieved the main tools and TTP's needed for introducing the operation actions on objectives and the intended effects (data exfiltration, defacement, data alteration and destruction, malware introduction, espionage and sabotage) are being brought in, be it another framework implant or living of the land utilities. Complex environments require complex tradecraft and higher level of operational security, since defenders focus is shifting from the lower levels of the David Bianco's Pyramid of Pain, IP, Hashes, Domains towards hunting tools and TTP's. · · · · · ·

REFERENCES

[1] S. Snape, "Introducing PoshC2 v8.0," LRQA Cyber Labs, 2022. [Online]. Available: https://www.lrqa.com/en/cyber-labs/introducing-poshc2-v8-0/. [Accessed: March 29, 2026].

[2] Bishop Fox, "Sliver: A cross-platform adversary emulation framework," GitHub, 2023. [Online]. Available: https://github.com/BishopFox/sliver. [Accessed: March 30, 2026]..

[3] Microsoft, "How AMSI helps (Antimalware Scan Interface)," Microsoft Learn, 2023. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/amsi/how-amsi-helps. [Accessed: Apr. 04, 2026].

[4] Microsoft, "PowerShell security features: Logging and transcription," Microsoft Learn, 2023. [Online]. Available: https://learn.microsoft.com/en-us/powershell/scripting/security/security-features. [Accessed: Apr. 02, 2026].

[5] D. Bianco, "The pyramid of pain," Enterprise Detection & Response, Mar. 2013. [Online]. Available: https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html. [Accessed: Apr. 02, 2026].

[6] Kaspersky, "Vulnerability landscape in Q4 2025," Securelist, Mar. 2026. [Online]. Available: https://securelist.com/vulnerabilities-and-exploits-in-q4-2025/119105/. [Accessed: March 29, 2026].

[7] The DFIR Report, "Threat actors' toolkit: Leveraging Sliver, PoshC2 & batch scripts," The DFIR Report, Aug. 2024. [Online]. Available: https://thedfirreport.com/2024/08/12/threat-actors-toolkit-leveraging-sliver-poshc2-batch-scripts/. [Accessed: Apr. 05, 2026].

[8] The MITRE Corporation, "MITRE ATT&CK," MITRE ATT&CK, 2025. [Online]. Available: https://attack.mitre.org/. [Accessed: Apr. 02, 2026].

[9] Fraunhofer FKIE, "APT29 - Threat actor profile," Malpedia, 2024. [Online]. Available: https://malpedia.caad.fkie.fraunhofer.de/actor/apt29. [Accessed: Apr. 05, 2026].

[10] Kaspersky, "Vulnerability landscape analysis for Q2 2025," Securelist, Aug. 2025. [Online]. Available: https://securelist.com/vulnerabilities-and-exploits-in-q2-2025/117333/. [Accessed: Apr. 05, 2026].

[11] Kaspersky, "Analyzing the vulnerability landscape in Q3 2025," Securelist, 2026. [Online]. Available: https://securelist.com/vulnerabilities-and-exploits-in-q3-2025/118197/. [Accessed: Apr. 06, 2026].

[12] Mandiant, "M-Trends 2025: Data, insights, and recommendations from the frontlines," Google Cloud Threat Intelligence, Apr. 2025. [Online]. Available: https://cloud.google.com/blog/topics/threat-intelligence/m-trends-2025. [Accessed: March 30, 2026].

[13] Help Net Security, "Attackers are handing off access in 22 seconds, Mandiant finds (M-Trends 2026)," Help Net Security, Mar. 2026. [Online]. Available: https://www.helpnetsecurity.com/2026/03/24/mandiant-m-trends-2026-report/. [Accessed: Apr. 05, 2026].

[14] Red Canary, "C2 frameworks," Red Canary Threat Detection Report, 2025. [Online]. Available: https://redcanary.com/threat-detection-report/trends/c2-frameworks/. [Accessed: Apr. 05, 2026].