SECURITY ADVISORY

August, 2025

PS1Bot Loader: Malvertising Meets Memory

Context

In August 2025, researchers uncovered a sophisticated malvertising campaign distributing the PS1Bot loader through search engine ads and compromised ad networks. Unsuspecting users searching for popular software were diverted to attacker-controlled domains hosting trojanized installers that mimicked legitimate applications.

Once executed, these installers trigger a multi-stage, in-memory infection chain designed to remain invisible to traditional security controls. At its core, the PS1Bot loader employs heavily obfuscated PowerShell and a modular payload delivery mechanism capable of deploying information stealers, remote access trojans (RATs), or ransomware on demand.

This campaign exemplifies the broader adversarial trend of abusing living-off-the-land binaries (LOLBins) such as PowerShell and Windows Installer, combined with social engineering through malvertising and SEO poisoning. By avoiding disk artefacts and executing entirely in memory, PS1Bot significantly complicates forensic analysis, impedes signature-based detection, and highlights the growing inadequacy of conventional antivirus solutions against modern, modular malware ecosystems.

Mitigation

Defending against fileless PS1Bot-like Loaders requires a proactive, layered security approach.

RestrHarden PowerShell & LOLBins – Enforce constrained language mode, block script execution from user-writable paths, and log high-risk commands (IEX, Add-Type).

Ad & Web Filtering – Deploy enterprise DNS and secure web gateways to block malvertising/SEO-poisoned domains and prevent access to attacker-controlled download sites.

Behavioural Monitoring & Response – Prioritise telemetry that detects in-memory execution, persistence in %ProgramData%/Startup, and suspicious Base64 exfiltration over traditional file signatures.

Technical Details

PS1Bot is a PowerShell-based loader designed as the orchestration layer of a multi-stage attack chain. Unlike commodity malware, it is engineered for fileless operation, executing entirely in memory and leaving minimal forensic artefacts. Its modular architecture enables on-demand delivery of secondary payloads—ranging from credential stealers to ransomware—while preserving operational stealth.

Infection and Delivery Mechanism

Malvertising & SEO Poisoning Explained

- **Malvertising:** The use of malicious or compromised online ads to redirect users to attacker-controlled sites or downloads. Ads appear legitimate, making them highly deceptive.

- **SEO Poisoning:** Manipulating search engine rankings so malicious sites or files appear at the top of search results, tricking users into clicking.

Malvertising & SEO Poisoning

The campaign leverages malvertising and SEO manipulation to funnel victims toward compressed archives (ZIP files) disguised as legitimate downloads. Filenames are engineered to mirror high-volume search queries, exploiting user trust in search engines and the perceived legitimacy of ads. Examples observed include:

- chapter 8 medicare benefit policy manual.zip
- Counting Canadian Money Worksheets Pdf.zip.e49
- zebra gx430t manual.zip.081
- kosher food list pdf (1).zip.c9a
- pambu panchangam 2024 25 pdf.zip.a7a

This delivery vector is particularly dangerous for enterprises because it bypasses traditional email-based security controls and instead exploits users during everyday search activity.

Attack Chain Breakdown

1. Malvertising Redirect

Victims encounter malicious ads injected into search results for popular tools and documents. Clicking the ad redirects them to attacker-controlled domains hosting trojanized MSI/EXE installers. This method sidesteps email gateways and exploits user trust in search engines.

2. Trojanized Installer Execution (Stage 1 Loader)

The fake installer shows benign UI elements while silently executing hidden PowerShell commands. These are often launched with powershell.exe -EncodedCommand, concealing intent. Payloads are staged entirely in memory, using Windows API calls like VirtualAlloc, WriteProcessMemory, and CreateRemoteThread for reflective injection.

3. PS1Bot Core Loader (Stage 2)

An obfuscated PowerShell script acts as the core loader. Persistence is achieved via:

- Registry Run keys (HKCU\Software\Microsoft\Windows\CurrentVersion\Run)
- Scheduled tasks executing encoded PowerShell at logon/startup

The loader performs anti-analysis checks for VMs, debuggers, and sandboxes, aborting if detected. It then establishes a secure HTTPS connection to C2 servers for modular tasking.

4. Payload Deployment (Stage 3 & 4)

PS1Bot retrieves encrypted payloads (often AES or RC4) from its C2, decrypting and injecting them directly into memory. Observed payloads include:

Info stealers (RedLine, Vidar) to capture credentials, browser data, crypto wallets

- Remote Access Trojans (RATs) for command execution and lateral movement
- Ransomware, deployed selectively based on operator intent
- Execution is maintained filelessly through reflective DLL injection or chained PowerShell loaders, ensuring stealth and operational flexibility.

Tactics, Techniques and Procedures

The following TTPs have been observed in relation to PS1Bot :

Tactic	Technique	ID	Details
Initial Access	Phishing: Spearphishing Attachment	T1566.001	Tax-themed phishing emails deliver ZIP archives containing malicious LNK shortcut files.
Execution	Command and Scripting Interpreter: PowerShell	T1059.001	Obfuscated PowerShell (`24.ps1`) executes shellcode to load Remcos RAT directly in memory.
Execution	Signed Binary Proxy Execution: MSHTA	T1218.007	`mshta.exe` launches obfuscated HTA files to continue the infection chain silently.
Persistence	Boot/Logon Autostart Execution: Registry	T1547.001	Registry key (`Rmc-7SY4AX`) stores persistence data; mutex ensures single active infection.
Defense Evasion	Process Injection	T1055	Shellcode injected into memory via APIs (e.g., `CallWindowProc`) to avoid disk-based detection.
Defense Evasion	Obfuscated Files or Information	T1027	Heavily obfuscated PowerShell and VBScript conceal payloads and frustrate static analysis.
Defense Evasion	Virtualization/Sandbox Evasion	T1497	Execution halts if sandbox, VM, or forensic artefacts are detected.
C2	Application Layer Protocol: Web Protocols	T1071.001	HTTPS-based C2 communications with domains such as `readysteaurants[.]com`.
C2	Encrypted Channel	T1573.002	Configurations and commands encrypted in RCData; TLS provides additional channel protection.
Collection	Data from Local System	T1005	Captures keystrokes, clipboard data, screenshots, and host system metadata.
Collection	Screen Capture	T1113	Records screenshots and exfiltrates them for operator review.
Collection	Audio Capture	T1123	Engages microphone to capture audio when instructed.

Cyber Threat Intelligence

The PS1Bot campaign underscores the growing convergence of malvertising and file-less execution, highlighting a strategic shift in how adversaries gain and maintain access to enterprise environments. By abusing PowerShell and deploying payloads entirely in memory, operators achieve stealth, agility, and rapid adaptability, frustrating detection and incident response teams.

Attribution and Threat Actor Motivation

Attribution remains inconclusive, with no clear links to a named APT group. However, the campaign's characteristics strongly suggest alignment with Malware-as-a-Service (MaaS) operators:

- Monetization-driven delivery: modular payloads enable revenue diversification through credential theft, ransomware, or resale of access.
- Flexibility for affiliates: the loader can be repurposed by different criminal actors, supporting various monetization models without requiring custom development.
- Shared infrastructure indicators: overlaps with commodity malware ecosystems point to a criminal syndicate model rather than a single state-sponsored actor.

Tradecraft and Targeting

- Delivery Vector: Leveraging malvertising and SEO poisoning indicates a deliberate focus on scale and accessibility—targeting broad user populations rather than spearphishing high-value individuals.
- Victim Profile: While opportunistic in nature, the tactic is high-impact for enterprises, where a single compromised endpoint can provide lateral access into privileged systems.
- Payload Agility: The modular architecture allows adversaries to deploy info stealers, RATs, or ransomware based on the victim's profile, financial value, or intelligence worth.
- Stealth Factors: Fileless execution ensures minimal forensic artefacts, complicating detection, attribution, and post-compromise investigations.

Strategic Implications

The campaign illustrates how search engines and online advertising ecosystems are now weaponized at scale, no longer limited to distributing potentially unwanted applications (PUAs). Instead, they have become reliable initial access vectors for advanced, modular malware frameworks. For large enterprises, this raises the stakes:

- Traditional email-centric security controls cannot mitigate malvertising.
- Incident response timelines shrink, as payloads never touch disk and pivot rapidly in memory.
- The threat model shifts toward continuous adversary presence, where actors can pivot from theft to extortion to espionage depending on evolving objectives.

PS1Bot PowerShell Loader (behavioural, lower FP)

High-confidence behavioural signature for the PS1Bot PowerShell loader (fileless/in-memory traits).

```
rule PS1Bot_PowerShell_Loader_InMemory_v1
{
 meta:
   author      = "CyberStash CTI"
   created     = "2025-08-19"
   version     = "1.0"
   description = "Detects PS1Bot-like PowerShell loaders that compile C# and inject shellcode in-memory"
   reference_1 = "Cisco Talos PS1Bot campaign (2025-08)"
   tlp         = "GREEN"

 strings:
   // PowerShell + C# interop patterns frequently seen in PS1Bot-style loaders
   $ps_addtype     = /Add-Type\s+-.*?/
   $interop_ns     = "System.Runtime.InteropServices"
   $dllimport      = "[DllImport(\"kernel32.dll\""
   // API set typically used for reflective injection
   $api_va         = "VirtualAlloc"
   $api_wpm        = "WriteProcessMemory"
   $api_crt        = "CreateRemoteThread"
   // Typical PowerShell loader constructs
   $from_b64       = "FromBase64String"
   $iex            = /Invoke-Expression|IEX/
   // Encoded blob markers occasionally present
   $b64_hint       = /[A-Za-z0-9+\/]{200,}={0,2}/

 condition:
   uint16(0) == 0xFEFF or filesize < 800KB and
   // Require C# interop + at least two injection APIs + PS execution traits
   1 of ($ps_addtype,$interop_ns,$dllimport) and
   2 of ($api_va,$api_wpm,$api_crt) and
   1 of ($from_b64,$iex,$b64_hint)
}
```

Malvertising JS Dropper (FULL DOCUMENT.js pattern)

Detector for the JS dropper stage often delivered via malvertising.

```
rule PS1Bot_JS_Dropper_v1
{
  meta:
    author      = "CyberStash CTI"
    created     = "2025-08-19"
    version     = "1.0"
    description = "Detects JS/VBScript-based droppers that spawn encoded PowerShell from malvertising ZIPs"
    reference_1 = "Talos/Infosecurity reporting on PS1Bot stage-1"
    tlp         = "GREEN"

  strings:
    $activex       = /ActiveXObject\(\s*\"WScript\.Shell\"/ nocase
    $spawn_ps_enc  = /powershell(\.exe)?\s+-enc(odedcommand)?\b/i
    $wscript       = /WScript\.(CreateObject|Shell)/i
    $filename_hint  = "FULL DOCUMENT.js"

  condition:
    filesize < 512KB and
    ( $spawn_ps_enc and ( $activex or $wscript ) ) or
    ( $spawn_ps_enc and $filename_hint )
}
```

Malvertising JS Dropper (FULL DOCUMENT.js pattern)

Detector for HTA/VBScript stage that silently chains into PowerShell.

```
rule PS1Bot_HTA_Stage_v1
{
 meta:
   author      = "CyberStash CTI"
   created     = "2025-08-19"
   version     = "1.0"
   description = "Flags HTA/VBScript stages that execute hidden PowerShell as part of the chain"
   tlp         = "GREEN"

 strings:
   $hta_sig        = "<HTA:APPLICATION" nocase
   $vbs_shell      = /CreateObject\(\s*\"WScript\.Shell\"/ nocase
   $run_hidden     = /Run\(\"powershell(\.exe)?\s+-enc(odedcommand)?/ nocase
   $mshta_comment  = "mshta chaining to PowerShell"

 condition:
   filesize < 400KB and
   ( $hta_sig or $vbs_shell ) and $run_hidden
}
```

Optional IOC Corroboration (Remcos/loader toolchain)

Optional/IOC-based: Very high-confidence hits tied to observed Remcos/ PS1Bot toolchain artefacts (use to confirm, not as the only control). Use as confirmatory hits in DFIR/retrohunt; these are strong but more specific to reported samples.

```
rule PS1Bot_Toolchain_Remcos_IOCs_v1
{
  meta:
    author      = "CyberStash CTI"
    created     = "2025-08-19"
    version     = "1.0"
    description = "Known artefacts from related PS1Bot/Remcos chains; high confidence when present"
    reference_1 = "Qualys, Infosecurity, HackRead 2025 reporting"
    tlp         = "AMBER"

  strings:
    $f_24ps1      = "24.ps1"
    $hta_311      = "311.hta"
    $hta_xlab     = "xlab22.hta"
    $mutex        = "Rmc-7SY4AX"
    $c2_domain    = "readysteaurants.com"

  condition:
    2 of them
}
```

References

IOCs:

https://github.com/Cisco-Talos/IOCs/blob/main/2025/08/ps1bot-malvertising-campaign.txt

Public Intelligence:

- https://thehackernews.com/2025/08/new-ps1bot-malware-campaign-uses.html
- https://www.broadcom.com/support/security-center/protection-bulletin/new-malicious-campaign-delivering-ps1bot-malware
- https://www.infosecurity-magazine.com/news/malvertising-powershell-malware/
- https://blog.talosintelligence.com/ps1bot-malvertising-campaign/