INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-16 19:17 UTC

FishMonger Deploys SprySOCKS Windows Variant with Kernel-Level Evasion Against Government Targets

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0484
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Windows systems (kernel driver abuse); government-sector targets in Honduras, Taiwan, Thailand, and Pakistan
Published	2026-06-16T14:02:57
Discovery Source	Rss

Executive Summary

FishMonger, a China-linked state espionage group, has deployed a new Windows version of its SprySOCKS backdoor against government entities in Honduras, Taiwan, Thailand, and Pakistan. The malware exploits kernel-level drivers to bypass modern endpoint security tools, giving attackers persistent, hidden access to compromised systems. Organizations supporting or partnering with government agencies in these regions, or those operating in sectors of strategic interest to China, face elevated targeting risk.

Technical Analysis

FishMonger has expanded the SprySOCKS implant, previously documented as Linux-only, to a Windows variant that abuses signed but vulnerable kernel drivers to achieve ring-0 code execution. This aligns with Bring Your Own Vulnerable Driver (BYOVD) tradecraft (MITRE T1068, T1574.008), in which a threat actor loads a legitimately signed, known-vulnerable driver to bypass Driver Signature Enforcement and execute code at the kernel level, defeating user-mode EDR hooks and visibility. The backdoor provides command-and-control capability (T1071), ingress tooling (T1105), rootkit-level persistence via kernel manipulation (T1014), and service-based persistence (T1543.003). Obfuscation techniques are in use (T1027). Infrastructure was likely pre-positioned using acquired resources (T1583.006). Applicable weaknesses are CWE-284 (Improper Access Control) and CWE-269 (Improper Privilege Management). No CVE identifiers have been publicly associated with this campaign. No vendor patch is available for the campaign itself; mitigation relies on driver blocklisting and hardening. Primary source is a T3 Dark Reading article; technical claims warrant validation against primary

research (e.g., ESET or Recorded Future publications on FishMonger).

Action Checklist

1. Containment, Enable and enforce Microsoft's recommended driver block rules via App Control for Business (formerly WDAC) on all Windows endpoints and servers, prioritizing systems with government or sensitive-sector data. Reference: https://learn.microsoft.com/en-us/windows/security/application-security/application-control/app-control-for-business/design/microsoft-recommended-driver-block-rules (T1 source, verified). Isolate any system exhibiting unsigned or anomalous kernel driver load events.
2. Detection, Hunt for BYOVD indicators: query Windows Security event logs for Event ID 7045 (new service installed) and Event IDs 6 and 219 (driver load failures or blocked drivers) in Microsoft-Windows-Kernel-PnP. Search Sysmon Event ID 6 (driver loaded) for drivers not present in your approved baseline. Flag any driver loads where the signing certificate matches known-vulnerable driver lists. Correlate with outbound C2 patterns on non-standard ports (T1071). No confirmed IOC hashes are available from the T3 source at time of reporting.
3. Eradication, Apply Microsoft's vulnerable driver blocklist via WDAC policy deployment. Audit all kernel drivers currently loaded across the environment against the Microsoft recommended driver block rules list (authoritative) and community resources such as loldrivers.io (supplementary reference only; validate all entries against the official Microsoft blocklist before removal). Remove or quarantine any driver matching known-vulnerable signatures. Review and remove any unauthorized services installed via T1543.003 tradecraft (unexpected services running from temp or user-writable directories).
4. Recovery, After driver remediation, validate EDR kernel sensor integrity: confirm EDR minifilter drivers are loading correctly and telemetry is flowing. Re-baseline kernel driver inventory on remediated hosts. Monitor for rootkit-level behavioral detections (MITRE T1014) for 30 days post-remediation. Implement NIST SI-4 (system monitoring) continuous monitoring controls to detect re-compromise. Align with CIS 7.3 to ensure automated OS patch management is current across the affected fleet.
5. Post-Incident, Conduct a gap assessment against NIST AC-6 (Least Privilege) and CIS 5.4 (Restrict Administrator Privileges) to determine whether kernel driver loading was possible due to overly permissive local admin rights. Review NIST CM controls for configuration management of approved driver baselines. Evaluate whether kernel-level logging and integrity monitoring (NIST AU-12, SI-7) are implemented across your monitoring stack. Brief leadership on BYOVD as a persistent tradecraft pattern requiring ongoing driver hygiene, not a one-time remediation.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate immediately to senior IR leadership and legal/privacy counsel if memory forensics or log analysis confirms SprySOCKS backdoor presence on any host processing government-classified, PII, or defense-related data, or if outbound C2 traffic indicates active data exfiltration — FishMonger's state-espionage mandate and the government-sector targeting profile of this campaign create mandatory breach notification obligations in multiple jurisdictions.

Recovery Notes	After WDAC policy enforcement and malicious driver/service removal, re-image any host where RAM forensics confirmed active SprySOCKS execution rather than attempting in-place recovery — FishMonger's kernel-level persistence mechanisms make confident eradication without reimaging unreliable. Post-reimage, re-baseline the full kernel driver inventory using `driverquery /FO CSV /SI` and store the output as a signed reference artifact. Maintain elevated Sysmon driver-load monitoring (Event ID 6) and weekly driver hash comparisons against the approved baseline for a minimum of 90 days, given FishMonger's demonstrated pattern of re-targeting previously compromised government networks.
Forensic Artifacts	RAM image (WinPmem or Magnet RAM Capture) from any host with anomalous driver loads — SprySOCKS operates in kernel space and its backdoor logic, injected shellcode, and active C2 socket state exist only in volatile memory and are not recoverable from disk after reboot or isolation | Windows System Event Log (evtx) filtered for Event ID 7045 (new service installed) — SprySOCKS installs its kernel driver as a Windows service; this event records the service name, binary path, and account context at installation time and is the primary host-based indicator of BYOVD tradecraft | Sysmon operational log Event ID 6 (driver loaded) entries showing ImageLoaded path, Hashes, and Signed/SignatureStatus fields — FishMonger's BYOVD technique loads a known-vulnerable legitimately signed driver alongside the malicious payload; the hash of the vulnerable driver will match loldrivers.io entries even when the SprySOCKS payload itself is novel | HKLM\SYSTEM\CurrentControlSet\Services registry hive export — records all driver and service registrations including those created by SprySOCKS for persistence; compare ImagePath values against known-good baselines, flagging entries pointing to non-standard paths such as %TEMP%, %APPDATA%, or user-writable directories | Network traffic PCAP from egress points covering the 72-hour window around initial detection — SprySOCKS is a backdoor with active C2 capability; packet captures will reveal beaconing intervals, protocol characteristics, and destination IPs/domains that constitute FishMonger infrastructure indicators for threat intelligence sharing and future detection rule development

Per-Action IR Details

Containment — Enable and enforce Microsoft's recommended driver block rules via App Control for Business (formerly WDAC) on all Windows endpoints and servers, prioritizing systems with government or sensitive-sector data. Reference: https://learn.microsoft.com/en-us/windows/security/application-security/application-control/app-control-for-business/design/microsoft-recommended-driver-block-rules (T1 source, verified). Isolate any system exhibiting unsigned or anomalous kernel driver load events.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: NIST AC-3 (Access Enforcement), NIST CM-7 — no mapped control (CM-7 not present in knowledge base; omitted), CIS 4.4 (Implement and Manage a Firewall on Servers)

Compensating: On hosts without WDAC licensing, use the free Windows Defender Application Control audit mode via PowerShell: `Set-CIPolicy -FilePath .\AuditPolicy.xml -Rules (New-CIPolicyRule -DriverFilePath -Level Hash)`. Deploy Sysmon with a configuration that logs Event ID 6 (ImageLoad) filtered to kernel-space drivers. Cross-reference loaded driver hashes against the loldrivers.io CSV export using a simple PowerShell loop: `Get-Content loldrivers.csv | Where-Object { $_ -match (Get-AuthenticodeSignature 'C:\Windows\System32\drivers\*.sys').SignerCertificate.Thumbprint }`. A 2-person team can triage the highest-risk hosts (internet-facing, government-data holders) within one shift.

Evidence: Before isolating any host showing anomalous driver loads, capture: (1) full RAM image using WinPmem or Magnet RAM Capture to preserve in-memory SprySOCKS backdoor artifacts and injected shellcode that will not survive reboot or isolation; (2) output of `fltMC` to enumerate active minifilter drivers — SprySOCKS leverages kernel drivers to blind EDR minifilters, so this baseline is critical; (3) output of `sc query type= driver state= all` and

`driverquery /FO CSV /SI` to document all loaded drivers with signing status before WDAC enforcement alters load behavior; (4) live network connections via `netstat -ano` and `Get-NetTCPConnection` to capture any active FishMonger C2 channels prior to host isolation.

Detection — Hunt for BYOVD indicators: query Windows Security event logs for Event ID 7045 (new service installed) and Event IDs 6 and 219 (driver load failures or blocked drivers) in Microsoft-Windows-Kernel-PnP. Search Sysmon Event ID 6 (driver loaded) for drivers not present in your approved baseline. Flag any driver loads where the signing certificate matches known-vulnerable driver lists. Correlate with outbound C2 patterns on non-standard ports (T1071). No confirmed IOC hashes are available from the T3 source at time of reporting.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

Controls: NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM, run the following PowerShell one-liner across endpoints to extract Event ID 7045 entries from the past 30 days: `Get-WinEvent -LogName System | Where-Object { $_.Id -eq 7045 } | Select-Object TimeCreated, Message | Export-Csv drivers_installed.csv`. For Sysmon Event ID 6 baselining, use Sigma rule 'sysmon_driver_load_win_vuln_drivers.yml' (available in the SigmaHQ repository) converted to a PowerShell query via sigma-cli. For C2 correlation without a SIEM, run Wireshark or tcpdump captures on egress points filtering for non-standard high ports (>1024, non-HTTP/HTTPS) with long-duration low-bandwidth sessions characteristic of SprySOCKS beaconing behavior.

Evidence: This is a detection/analysis step that does not alter live state; no volatile pre-capture is required before running queries. However, preserve all raw log exports immediately upon collection — SprySOCKS's kernel-level evasion may allow FishMonger to tamper with or clear logs during an active intrusion. Prioritize collecting: Windows System Event Log (evt/evtx) for Event ID 7045 entries; Microsoft-Windows-Kernel-PnP operational log for Event IDs 6 and 219; Sysmon operational log focusing on Event ID 6 (driver loads) and Event ID 3 (network connections) for C2 beacon patterns; and the HKLM\SYSTEM\CurrentControlSet\Services registry hive, which records all installed services and drivers including those SprySOCKS may install for persistence.

Eradication — Apply Microsoft's vulnerable driver blocklist via WDAC policy deployment. Audit all kernel drivers currently loaded across the environment against the Microsoft recommended driver block rules list and the loldrivers.io community database (treat loldrivers.io as a secondary reference; validate entries against the Microsoft blocklist). Remove or quarantine any driver matching known-vulnerable signatures. Review and remove any unauthorized services installed via T1543.003 tradecraft (unexpected services running from temp or user-writable directories).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST SI-2 (Flaw Remediation), CIS 2.3 (Address Unauthorized Software), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: For teams without enterprise patch management, use `Get-WindowsDriver -Online | Export-Csv all_drivers.csv` to enumerate installed drivers, then compare hashes against the Microsoft WDAC blocklist XML using a PowerShell diff script. For unauthorized service removal tied to SprySOCKS persistence (services launched from %TEMP%, %APPDATA%, or user-writable paths), run: `Get-WmiObject Win32_Service | Where-Object { $_.PathName -match 'Temp|AppData|Users' } | Select-Object Name, PathName, StartMode | Export-Csv suspicious_services.csv`. Quarantine identified binaries to an isolated folder before deletion to preserve forensic copies. Use ClamAV with a custom YARA rule targeting SprySOCKS behavioral patterns (socket creation + kernel driver load sequence) to scan quarantine artifacts.

Evidence: Before removing any driver or service associated with SprySOCKS, capture: (1) full RAM image if not already acquired — eradication actions will destroy in-memory indicators of the backdoor's current operational state and any injected code; (2) a forensic copy of the suspicious driver file(s) from disk (typically in

%SystemRoot%\System32\drivers\ or user-writable paths) using `robocopy` with /COPYALL flags to preserve metadata and timestamps; (3) registry export of HKLM\SYSTEM\CurrentControlSet\Services for all service/driver entries before deletion — FishMonger may use multiple persistence keys; (4) `sc qc ` output for each suspicious service to document binary path, start type, and account context before removal; (5) parent process tree from Sysmon Event ID 1 (Process Create) showing what process originally installed the malicious service.

Recovery — After driver remediation, validate EDR kernel sensor integrity: confirm EDR minifilter drivers are loading correctly and telemetry is flowing. Re-baseline kernel driver inventory on remediated hosts. Monitor MITRE T1014 (rootkit) behavioral detections for 30 days post-remediation. Implement NIST SI-4 (system monitoring) continuous monitoring controls to detect re-compromise. Align with CIS 7.3 to ensure automated OS patch management is current across the affected fleet.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-12 (Audit Record Generation), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 8.2 (Collect Audit Logs)

Compensating: Without commercial EDR, verify minifilter driver health using `fltMC` — all expected security tool filters should appear with a valid altitude and no error state. Re-baseline the approved driver inventory by running `driverquery /FO CSV /SI > baseline_post_remediation.csv` and storing it as the new signed reference. For ongoing rootkit behavioral monitoring, deploy or update Sysmon with Event ID 6 alerting on any new driver load not matching the post-remediation baseline hash list. Schedule a weekly cron/Task Scheduler job to re-run the driver hash comparison and alert on deviations, substituting for continuous EDR telemetry.

Evidence: Recovery steps alter system state (restoring services, validating configurations); before any recovery action on a host, confirm the prior volatile capture from the eradication phase is complete and stored to write-protected media. During recovery validation, collect: (1) `fltMC` output post-recovery to confirm EDR minifilter altitude and load status — SprySOCKS specifically targets minifilter drivers to blind EDR, so their restored presence is a primary recovery health indicator; (2) re-run `Get-NetTCPConnection` and review DNS query logs (Microsoft-Windows-DNS-Client operational log) for 72 hours post-recovery to confirm FishMonger C2 beaconing has ceased; (3) Windows Security Event Log Event ID 4688 (Process Creation) for any re-emergence of processes associated with the SprySOCKS loader or its parent service.

Post-Incident — Conduct a gap assessment against NIST AC-6 (Least Privilege) and CIS 5.4 (Restrict Administrator Privileges) to determine whether kernel driver loading was possible due to overly permissive local admin rights. Review NIST CM controls for configuration management of approved driver baselines. Evaluate whether D3-SFA (System File Analysis) and D3-SICA (System Init Config Analysis) countermeasures are implemented in your monitoring stack. Brief leadership on BYOVD as a persistent tradecraft pattern requiring ongoing driver hygiene, not a one-time remediation.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: NIST AC-6 (Least Privilege), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: For teams without a GRC platform, conduct the privilege gap assessment manually: run `net localgroup administrators` on all affected hosts and export results. Cross-reference against CIS 5.4 criteria — any non-IT account with local admin rights on a government-data system is a finding. Document the BYOVD-specific gap: kernel driver loading via `sc create` or `NtLoadDriver` requires SeLoadDriverPrivilege, which should be restricted to SYSTEM and dedicated admin accounts only. Audit this via `whoami /priv` on service accounts and `secedit /export /cfg secpol.cfg` to review local security policy privilege assignments. Use this output as the basis for the leadership brief, framing SprySOCKS as a recurring FishMonger tradecraft pattern, not a one-time event.

Evidence: Post-incident activity does not alter active system state on remediated hosts; no new volatile capture is required at this phase. Reference the previously collected artifacts: (1) the registry export of HKLM\SYSTEM\CurrentControlSet\Services to document how the malicious driver service was registered and under

what account context — this directly informs the AC-6/CIS 5.4 gap assessment; (2) the Sysmon Event ID 1 process creation logs showing the privilege context of the SprySOCKS installer process; (3) Windows Security Event ID 4673 (Sensitive Privilege Use) logs for SeLoadDriverPrivilege invocations in the period preceding detection — this establishes the precise privilege path FishMonger exploited and quantifies the gap.

Detection Guidance

Primary detection surface is kernel driver load activity. Use Sysmon Event ID 6 to log all driver loads; alert on any driver whose SHA-256 hash matches the Microsoft recommended driver block list or community BYOVD driver databases. Monitor Windows Security Event ID 7045 for new service creation from non-standard paths (temp directories, user profiles, AppData). Alert on Event ID 219 (driver load error) combined with subsequent suspicious process activity, which may indicate a failed or probed BYOVD attempt. For network-layer indicators, flag beaconing patterns consistent with SOCKS-proxied C2 traffic (T1071): periodic, low-volume outbound connections to non-categorized external IPs, especially over ports 443 or 80 with non-browser user agents. Hunt for processes injecting into or spawning from kernel-adjacent services. Apply local account monitoring (NIST AU-12, MITRE T1014 behavioral detection) to detect privilege escalation sequences following driver load events. No confirmed IOC hashes, domains, or IPs are available from the current T3 source; monitor for primary research releases from vendors with direct FishMonger telemetry (ESET, Recorded Future) for hash-level IOCs.

Framework Mappings

MITRE-ATTACK

- **T1068** — Exploitation for Privilege Escalation
- **T1105** — Ingress Tool Transfer
- **T1071** — Application Layer Protocol
- **T1014** — Rootkit
- **T1543.003** — Windows Service
- **T1583.006** — Web Services
- **T1190** — Exploit Public-Facing Application
- **T1027** — Obfuscated Files or Information
- **T1574.008** — Path Interception by Search Order Hijacking

NIST-800-53R5

- **AC-6** — Least Privilege
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **CA-7** — Continuous Monitoring
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SI-7** — Software, Firmware, and Information Integrity

- **AC-3** — Access Enforcement

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control

CIS-V8

- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts
- **6.8** — Define and Maintain Role-Based Access Control
- **8.2** — Collect Audit Logs

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control

ISO-27001-2022

- **A.5.34** — Privacy and protection of personal information

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1068	Exploitation for Privilege Escalation	Privilege-Escalation
T1105	Ingress Tool Transfer	Command-And-Control
T1071	Application Layer Protocol	Command-And-Control
T1014	Rootkit	Defense-Evasion
T1543.003	Windows Service	Persistence
T1583.006	Web Services	Resource-Development
T1190	Exploit Public-Facing Application	Initial-Access
T1027	Obfuscated Files or Information	Defense-Evasion
T1574.008	Path Interception by Search Order Hijacking	Persistence

Sources

Source	URL	Tier
Security News	https://www.darkreading.com/threat-intelligence/sprysocks-windows-v...	T3
5 Zero-Days Found in Windows Kernel Drivers - LinkedIn	https://www.linkedin.com/posts/jehadabudagga_5-zero-days-in-under-2...	T3
Microsoft recommended driver block rules	https://learn.microsoft.com/en-us/windows/security/application-secu...	T1
Signed kernel drivers – Unguarded gateway to Windows' core	https://www.welivesecurity.com/2022/01/11/signed-kernel-drivers-ung...	T3
Hunting Vulnerable Kernel Drivers - VMware Security Blog	https://blogs.vmware.com/security/2023/10/hunting-vulnerable-kernel...	T3

Generated 2026-06-16 19:17 UTC by TJS Security Command Center