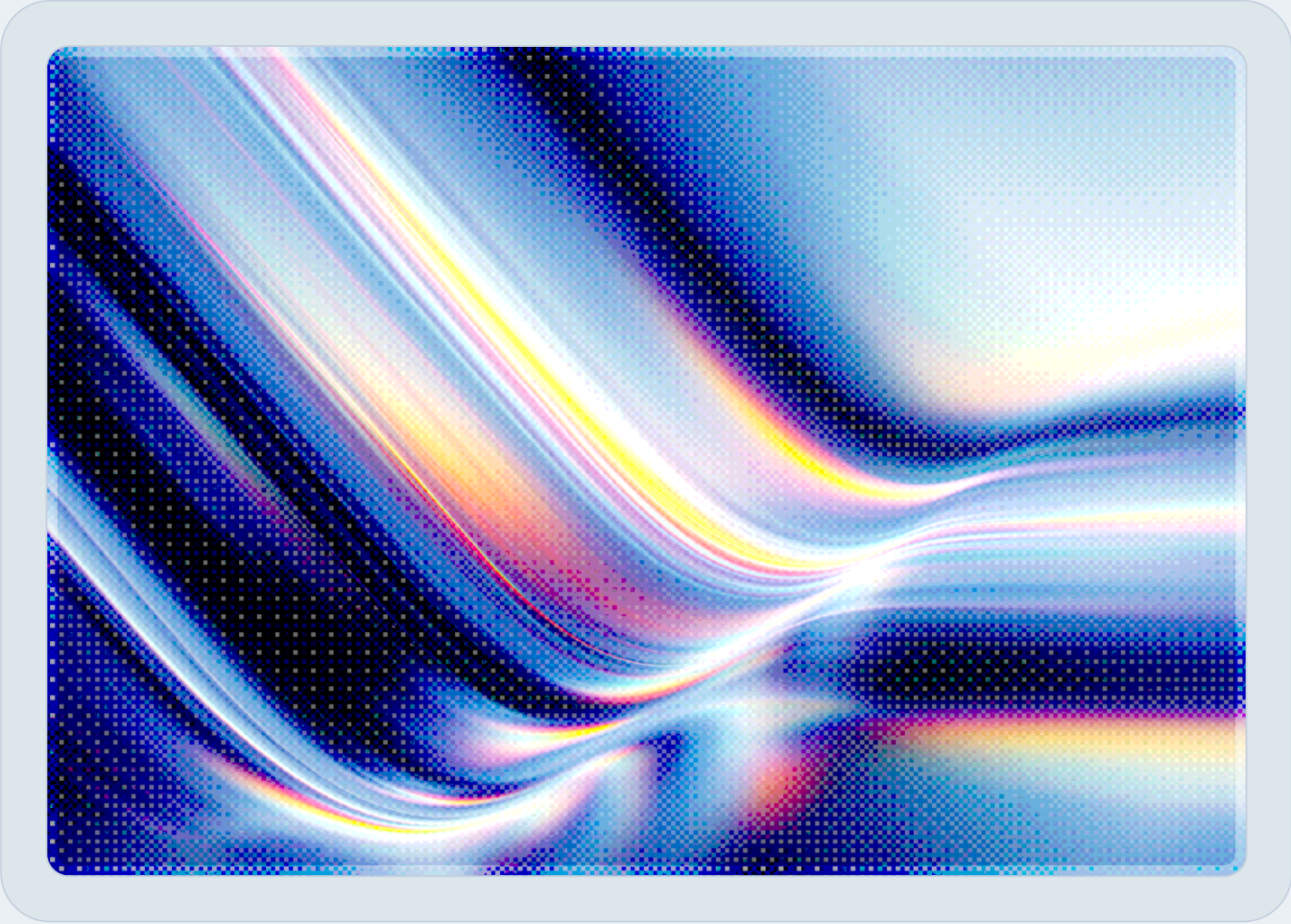
CSAI Foundation | Cloud Security Alliance

SLEEPWALKER: A Network-Triggered Backdoor With No Outbound C2

2026-08-27

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- SLEEPWALKER is a previously undocumented, unsigned 64-bit Windows DLL that impersonates Microsoft's `dpapi.dll` and side-loads into `ERAAgent.exe`, the ESET Management Agent process, where it remains dormant in memory until it receives a specially crafted network packet [1][2].
- The backdoor contains no hardcoded command-and-control domains, IP addresses, or URLs and never initiates an outbound connection on its own; it instead sniffs passing network traffic for a "magic packet" that must clear a length check, an XOR-derived validation step, a CRC-32 checksum, and AES-256-CCM authenticated decryption before it is treated as a command [3][1].
- Commands arrive as programs written in a proprietary 23-instruction bytecode language covering scheduling, TCP/UDP/ICMP/named-pipe data transfer, staged payload delivery with SHA-256 verification, decompression, and in-memory shellcode execution — a design independent researcher Dominik Reichel described as making the implant "hard to catch from the network side" because there is nothing to block until an operator sends the one packet that activates it [3][1].
- Persistence relies entirely on DLL search-order abuse of a legitimate, trusted security agent rather than on exploiting a software vulnerability; the malware activates only when its host process is named `ERAAgent.exe`, and writing the file in the first place requires local administrator access already obtained through some other means [2][4].
- Attribution and victim scope are unknown: the analysis rests on a single binary submitted with no collection context, no confirmed victims, and no code overlap with known threat-actor toolsets, though the sophistication of the covert-channel design is consistent with a targeted, well-resourced operation rather than commodity malware [1][2][3].

Background

On August 24, 2026, independent malware researcher Dominik Reichel, publishing under the handle R136a1, disclosed a previously unknown Windows backdoor he named SLEEPWALKER [3]. The sample is a 59,904-byte, unsigned 64-bit DLL compiled with a timestamp of June 10, 2024, that forges ESET Management Agent version metadata and exports the same seven functions as Microsoft's legitimate

Data Protection API library, `dpapi.dll` [3][2]. Rather than functioning as an initial-access tool, SLEEPWALKER is built to be dropped into the ESET Management Agent's installation directory by an attacker who already holds local administrator rights on the target host, where it loads automatically the next time the `ERAAgent.exe` service starts and stays resident in memory without writing further files, opening a listening port, or contacting any server on its own [1][3]. Trade press coverage of the finding, including reporting from The Hacker News and The Register, has since amplified the disclosure and reiterated Reichel's core assessment: the most-repeated point in subsequent coverage is that it does nothing detectable until it receives a single, precisely formatted network packet [1][2].

SLEEPWALKER can be grouped with a small number of "passive" or magic-packet-triggered backdoors that evade conventional network monitoring by design rather than by obfuscation [5][6]. The best-documented predecessor is BPFDoor, a Linux implant attributed to the China-nexus group Red Menshen (also tracked as DecisiveArchitect or Red Dev 18) that has been deployed against telecommunications, finance, and retail targets across South Korea, Hong Kong, Myanmar, Malaysia, and Egypt since at least 2021 [5][6]. Like SLEEPWALKER, BPFDoor uses Berkeley Packet Filter logic to inspect traffic for a magic value before waking up, allowing it to respond to a trigger packet even when a host firewall would otherwise block the port, and it does not require an open listening port or an outbound beacon to a command server [5]. In CSA's assessment, SLEEPWALKER's Windows implementation extends the same passive-trigger philosophy with a substantially more elaborate command language and a wider set of covert transports than BPFDoor — including one, VMware's Virtual Machine Communication Interface, that operates entirely outside conventional network capture between a guest and its host [3][7].

The choice of ESET's management agent as a host process also echoes a prior documented case of DLL-search-order abuse against ESET software. In April 2025, Kaspersky researchers disclosed that the China-linked APT group ToddyCat had exploited CVE-2024-11859, a DLL search-order hijacking flaw in ESET's command-line scanner, to load a malicious `version.dll` proxy and deploy malware Kaspersky named TCESB, which disabled Windows kernel-level security notifications [8]. SLEEPWALKER's mechanism differs in one important respect: Reichel's analysis found no indication that it exploits a patched vulnerability in ESET's software; instead, it relies on the ordinary Windows DLL search order and the fact that an attacker with administrator rights can simply place a like-named file in the agent's own directory [3][2]. In CSA's assessment, this distinction matters for defenders because there is no ESET patch that closes this particular avenue — the malware abuses trust in a legitimate, signed process rather than a flaw ESET can fix, which places the burden of detection on file-integrity and process-behavior monitoring rather than patch management.

Security Analysis

A trigger designed to leave nothing to block

SLEEPWALKER's behavior is consistent with a design goal of leaving no persistent artifact for network defenders to detect. The implant opens no listening socket by default and places no command-and-control infrastructure in the binary; Reichel's analysis found no domains, IP addresses, or URLs anywhere in the sample [1][3]. Instead, it installs a raw packet listener across all monitored network interfaces — including, notably, traffic destined for other machines when the host sits on a gateway or bridged network — and evaluates every observed packet against a strict validation sequence: a minimum 48-byte length, an XOR-derived length candidate computed from trailer bytes, a CRC-32 checksum, and finally AES-256-CCM authenticated decryption using a statically linked mbedTLS library [1][3]. Reichel reported that his analysis recovered the embedded AES key and configuration nonce directly from the binary, which is what made full reverse engineering of the trigger and command format possible; only a packet that clears every step in order is treated as a valid instruction, and anything else passes through unremarked [3]. A second trigger path — a DNS-based variant that encodes tasks in Base32 inside DNS query labels with CRC-8 markers — is fully implemented in the code but was not active in the analyzed sample, suggesting either a capability held in reserve or a configuration the operator had not yet deployed [3][2].

A purpose-built bytecode language, not a text-based command shell

Once a trigger packet validates, SLEEPWALKER does not execute a plain command string; it interprets an encrypted program written in a custom 23-instruction bytecode language whose opcodes, in Reichel's words, exist "nowhere but inside this one file" [1][3]. The instruction set groups into five functional categories: control flow (exiting or spawning a scripted thread), timing and scheduling (fixed or randomized sleep intervals, cron-style scheduling, looping and repetition), outbound data transmission over TCP, UDP, ICMP, or named pipes, inbound task reception over the same channels, and execution primitives that stage a file, verify it against a SHA-256 hash, decompress and run it, or execute shellcode directly in memory [3]. This design gives an operator the ability to compose multi-step, conditional operations — for example, waiting a randomized interval, then verifying a staged payload's hash before executing it — entirely within an opaque, encrypted blob that a defender cannot interpret without first recovering the AES key and independently reverse engineering the instruction set [3]. In CSA's assessment, that requirement exceeds what static signature matching or conventional sandbox emulation is typically able to clear.

Six covert channels, several purpose-built to avoid standard monitoring

Beyond the raw packet trigger, SLEEPWALKER supports at least six distinct communication mechanisms once active: TCP and UDP sockets, ICMP echo-request payloads (which conceal data inside ping traffic), SMB named pipes that can support credentialed lateral movement to other Windows systems on the same network, promiscuous-mode raw packet capture, and VMware VMCI, which carries traffic between a guest and its hypervisor host entirely outside the conventional network stack and is therefore invisible to network-based packet capture tools [3][7]. To enable unauthenticated named-pipe communication, the malware also modifies two Windows security settings — setting the `EveryoneIncludesAnonymous` registry value to 1 and adding entries to `NullSessionPipes` — changes that, in principle, are detectable at the host level even though the network channel itself is designed to be quiet [1][3]. Taken together, the diversity of transports is consistent with — though not confirmed by any source as evidence of — an operator preparing fallback channels in case any single control, such as perimeter monitoring, EDR telemetry, or hypervisor-boundary assumptions, were in place. No source states this was the actual design rationale.

Persistence through trust, not exploitation, and an unresolved attribution question

SLEEPWALKER's persistence model is narrow: the DLL checks only the host process's name, not its digital signature or file path, and remains inert unless it is loaded as `dpapi.dll` inside a process named `ERAAgent.exe` [3][2]. No UAC bypass or privilege-escalation logic is present in the sample, because none is needed — the malware assumes it is already running with whatever privileges the ESET Management Agent service itself holds, and the act of writing the file into that directory in the first place already required local administrator access [1][3]. This indicates SLEEPWALKER is a post-compromise implant intended to extend an existing foothold rather than an initial-access tool, which in turn means its presence on a host is itself evidence of a prior, unaddressed compromise. Reichel was explicit about the limits of what a single sample can establish: he found no code overlap with known threat-actor toolsets and no information about the initial access vector, deployment scope, or actual victims, cautioning that "the assessment rests on a single binary supplied with no collection context" [1][2]. That said, the combination of a bespoke instruction set, hardened trigger validation, and multiple fallback covert channels is not something commodity malware kits typically include, and the sources reviewed for this note — Reichel's original analysis and subsequent trade-press coverage — characterize the engineering effort as consistent with a targeted, well-resourced operation, though formal attribution is not yet possible [3][2].

Recommendations

Immediate Actions

Organizations running ESET Management Agent should treat the presence of a file named `dpapisvc.dll` – which has no legitimate reason to exist in this directory, since SLEEPWALKER forwards its proxied calls to that name – or any unexpected `dpapi.dll` alongside `ERAAgent.exe`, as a high-confidence compromise indicator, and should isolate the affected host for forensic review rather than simply deleting the file [1][3]. Security teams can use the SHA-256 hash `d347170752a28e2b8c4b8b9f3cab2e3a6541ba11682c94498d26eb9002779d60` and MD5 hash `2318327b29bb1c0e2d2b5f0211fc7fac` published with Reichel's analysis to scan endpoint file inventories, and should deploy the YARA rule he released, which targets the sample's static AES key material, configuration nonce, and trigger-validation logic [1][3]. Hosts should also be checked for the registry indicators associated with the implant's named-pipe manipulation – an `EveryoneIncludesAnonymous` value of 1 and unexpected entries in `NullSessionPipes` – since these changes persist independently of whether the DLL is currently loaded [1][3].

Short-Term Mitigations

Because SLEEPWALKER's persistence depends on Windows DLL search-order behavior rather than a patchable vulnerability, defenders should apply file-integrity monitoring and application allowlisting to security-agent installation directories specifically, flagging any DLL write into those paths that does not correspond to a signed vendor update. Endpoint detection tooling that alerts on unsigned DLLs loading into signed, trusted processes – a detection approach that does not depend on recognizing SLEEPWALKER's specific bytecode or network signature – would catch this and structurally similar implants regardless of variant [2][8]. Given that the malware is explicitly designed to leave no persistent network footprint, security teams should not rely on network monitoring alone to rule out compromise; process-level and file-system telemetry are the more reliable detection surface for this class of threat, consistent with the same lesson drawn from the ToddyCat/TCESB abuse of ESET's scanner roughly sixteen months earlier [8][3].

Strategic Considerations

SLEEPWALKER is the second publicly documented example – after BPFDoor on Linux – of a passive, magic-packet-triggered implant built to defeat security architectures that equate "no outbound connection" with "no compromise" [5][6]. Organizations should reassess detection strategies that lean

primarily on egress monitoring or command-and-control domain reputation, since both are structurally blind to implants of this design, and should extend the same scrutiny to any security or management agent capable of running with elevated privileges, since such agents are attractive side-loading targets precisely because their trusted status suppresses routine scrutiny. Given that a single publicly analyzed sample cannot establish deployment scope, security teams should also treat the disclosure as a prompt to audit their own environments for the specific indicators published, rather than waiting for confirmed in-the-wild sightings before acting.

CSA Resource Alignment

SLEEPWALKER's core design principle — a stealthy implant that avoids conventional network-based detection by suppressing outbound connections and blending into trusted infrastructure — parallels the dynamic CSA examined in VerdantBamboo Deploys BRICKSTORM BSD Variant on Linux Appliances [10], which documented a China-nexus actor achieving an 18-month undetected dwell time by deploying a backdoor on network-edge appliances specifically because such devices sit outside the coverage of typical endpoint and network monitoring tooling. That report's recommendation to extend centralized logging and EDR coverage to under-monitored infrastructure, rather than assuming perimeter and egress monitoring are sufficient, applies directly to SLEEPWALKER: an implant engineered to make zero outbound connections defeats exactly the detection model that report identified as a recurring blind spot, reinforcing the case for host- and process-level telemetry as the more durable control.

Organizations mapping this threat to a formal controls framework should reference the Threat and Vulnerability Management and Application and Interface Security domains of CSA's AI Controls Matrix (AICM) v1.1 [9], which provide control objectives for detecting unauthorized code execution and unsigned-binary loading into trusted processes — precisely the abuse pattern SLEEPWALKER exhibits by side-loading into ESET's signed management agent. The AICM's Identity and Access Management domain is also directly relevant given SLEEPWALKER's registry manipulation of anonymous named-pipe access (`EveryoneIncludesAnonymous`, `NullSessionPipes`), a host-level identity weakening that AICM's IAM controls are designed to flag through configuration baseline monitoring [9].

Finally, SLEEPWALKER's reliance on the implicit trust extended to a signed security agent process is a concrete illustration of the continuous-verification principle embedded in AICM's identity and workload-trust controls, which reject one-time trust decisions based on a binary's signature or its parent process's reputation [9]. This same logic underlies CSA's Software-Defined Perimeter and Zero Trust guidance, which describes continuous validation of identity, device, and context as a foundational Zero Trust principle rather than durable trust granted on the strength of a single signature or vendor-reputation check [11]. A Zero Trust posture that verifies process behavior and file provenance on an ongoing basis,

rather than granting durable trust to any process simply because it belongs to a recognized security vendor, would surface the unsigned `dpapi.dll` side-load that SLEEPWALKER depends on regardless of how quiet its network behavior remains.

References

[1] The Hacker News. "New SLEEPWALKER Backdoor Waits for One Crafted Packet, Then Runs Its Own Bytecode." The Hacker News, August 26, 2026.

[2] The Register. "You Don't Want This Sleepwalker Backdoor on Your Windows Machine." The Register, August 24, 2026.

[3] Dominik Reichel (R136a1). "SLEEPWALKER: A Passive Backdoor With Its Own Command Language." R136a1, August 24, 2026.

[4] GBHackers. "SLEEPWALKER Backdoor Uses Magic Packet, DLL Side-Loading and In-Memory Shellcode Execution." GBHackers, August 2026.

[5] The Hacker News. "China-Linked Red Menshen Uses Stealthy BPFDoor Implants to Spy via Telecom Networks." The Hacker News, March 2026.

[6] Trend Micro. "BPFDoor's Hidden Controller Used Against Asia, Middle East Targets." Trend Micro Research, 2025.

[7] Cyberpress. "SLEEPWALKER Backdoor Uses SMB, ICMP, DNS and VMware VMCI for Covert Communications." Cyberpress, August 2026.

[8] Securelist (Kaspersky). "APT Group ToddyCat Exploits a Vulnerability in ESET for DLL Proxying." Kaspersky Securelist, April 2025.

[9] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, 2026.

[10] Cloud Security Alliance. "VerdantBamboo Deploys BRICKSTORM BSD Variant on Linux Appliances." CSA AI Safety Initiative, June 8, 2026.

[11] Cloud Security Alliance. "Stealth Mode SDP for Zero Trust Network Infrastructure." Cloud Security Alliance, February 2026.

