halcyon

The Executive Guide to Ransomware Tactics

5 EDR Evasion Techniques Used by Ransomware Attackers

The Tradecraft of Evasion

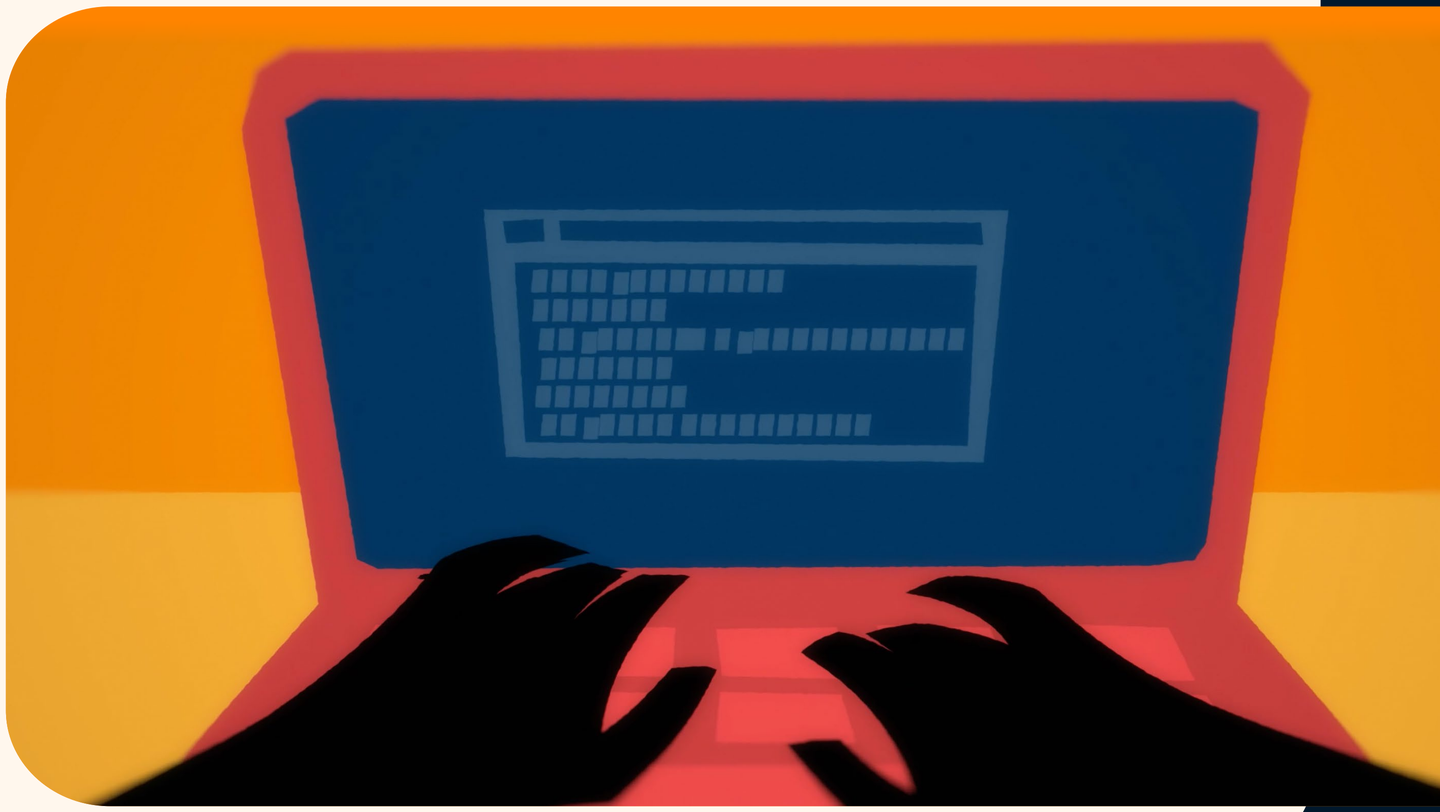
Practitioners who have been around security long enough all know there is no silver bullet when it comes to cybersecurity. But why is that? **EDR tools are essential, yet attackers consistently find ways to bypass them.** Finding malware on an endpoint may or may not be a high-alert situation, but ransomware almost always is. Ransomware can quickly escalate, turning a record quarter into bankruptcy and a real risk of permanently closing your doors.

The tradecraft that can go into EDR evasion techniques ranges from partially concealed to full-fledged clandestine. Meet the top five evasion techniques that turn shady tradecraft into full-blown ransomware drama (popcorn optional).

1 EDR Tampering: Lights Out for Endpoint Agents

Here's a bit of hard truth – threat actors already know you have an EDR. As a matter of fact, they count on it. Agent tampering or neutralization is at the top of any ransomware group's to-do list. Here are a few tried-and-true methods attackers use to force your endpoint agent to take the day off:

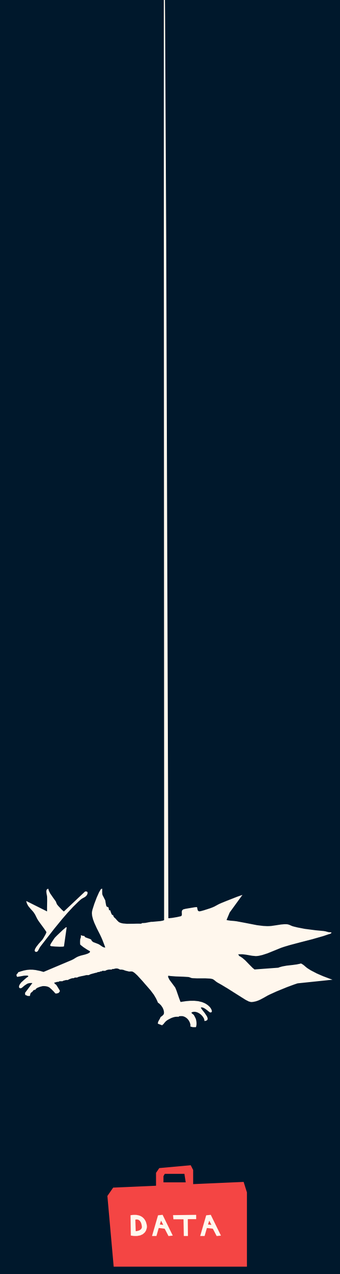
- Stopping or reconfiguring agent services
- Abusing stolen admin credentials to change protection settings
- Tampering with kernel drivers (more on this later)
- Blocking the agent's ability to report to its management console



The plan is simple: disable the alarm, giving attackers a leisurely shopping spree through your data to steal what they can, and lock the rest for posterity. Neutralizing endpoint agents means immediately removing primary telemetry and any automated responses you had in store. Whether your SOC team is internal or outsourced, they're now hunting in the dark or, worse yet, oblivious to the threat.

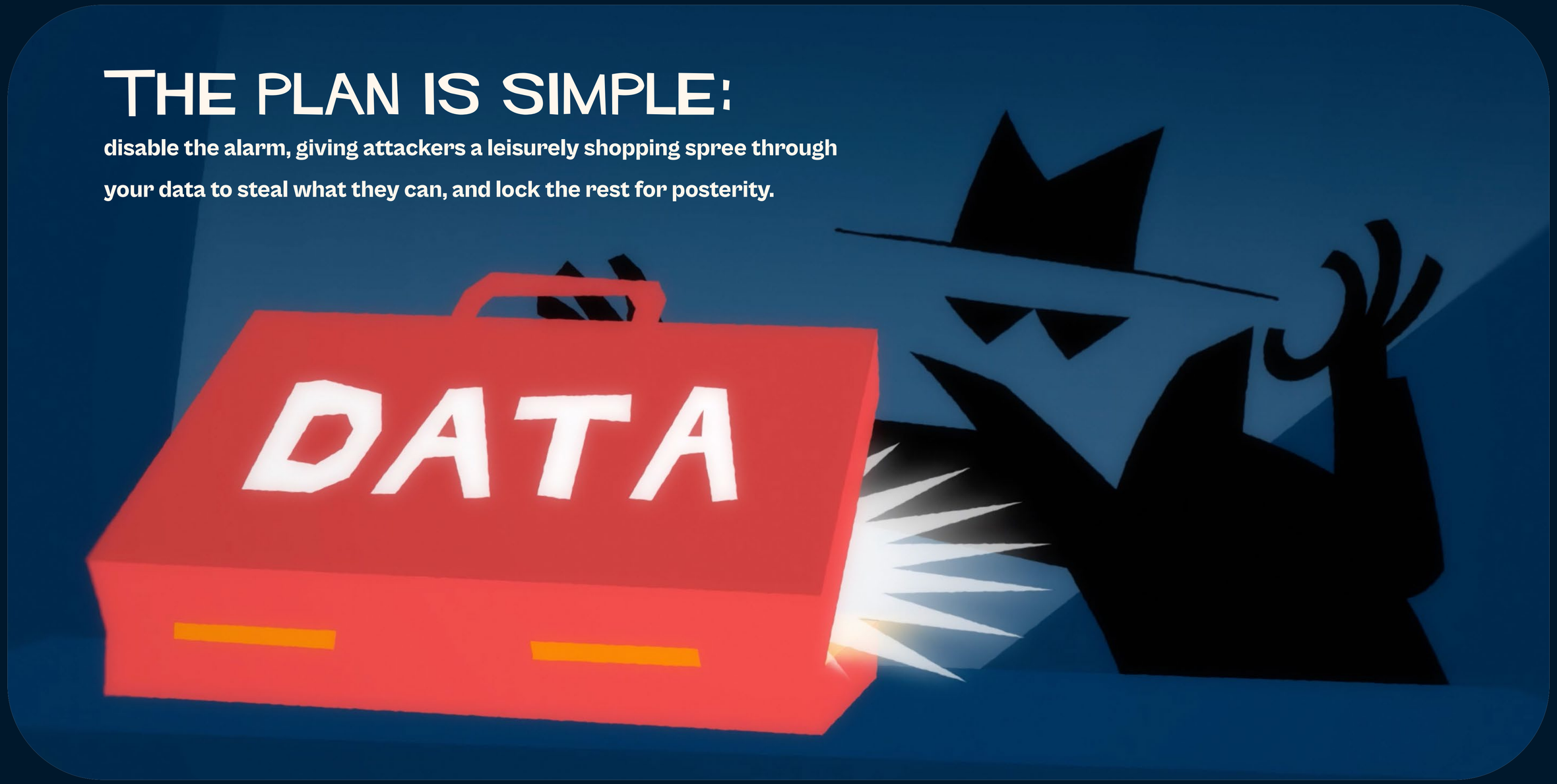
When things go quiet, it's rarely good news. Sudden heartbeat gaps, mass agent restarts, tamper-protection mysteriously toggled off, and a synchronized drop in event volume all spell trouble. Attackers don't even need a long window once the lights are out. Groups like Akira often encrypt only 1 to 2 percent of files, because that still creates the same operational downtime as encrypting everything. That means a single compromised machine with agent protections neutralized can quietly reach out and encrypt data across hundreds or thousands of systems in a very short window, long before anyone realizes the telemetry has gone dark.

The fix is less glamorous but far more effective: enforce vendor tamper-protection, lock down who can fiddle with agent configs, keep an eye on heartbeat and policy changes, and make sure your logs live somewhere attackers can't "edit for clarity."



THE PLAN IS SIMPLE:

disable the alarm, giving attackers a leisurely shopping spree through your data to steal what they can, and lock the rest for posterity.



2 BYOVD: KERNEL-LEVEL CHAOS

Bring Your Own Vulnerable Driver, or BYOVD, is the hacker express lane into the kernel. Just drop in a validly signed vulnerable driver, and suddenly you're running the show. **Drivers sit at the top of the privilege food chain, so this tactic gifts a skeleton key to the OS and one that opens doors better left sealed.** Gaining kernel read-write privileges allows the ability to inspect, adjust, or delete any runtime structure that EDR depends on. Simply said, this gives threat actors the ability to veto even admin-level commands. Here's a playlist of some of the top methods used by ransomware groups:

Callback Removal:
The equivalent of cutting power to motion sensors while leaving the lights on. Everything looks fine, but no monitoring or alarms are happening.

- **USER LEVEL:** Tamper with user-space APIs or agent components, so no events are forwarded.
- **KERNEL LEVEL:** Alters callback tables or driver hooks that provide system-wide visibility.
- **EDR INTERNALS:** Modifies the agent's telemetry pipeline or in-memory structures, so it doesn't report certain events.

Mini-filter Detachment:

Like quietly unbolting the security camera from the wall while leaving the housing in place. The system still thinks file operations are being watched, but the filter that sees and reports them is gone.

- **USER LEVEL:** Nudge the user-space pieces into silence, break the chain that forwards events from apps and agents, so nothing ever rings the bell.
- **KERNEL LEVEL:** Quietly unhook kernel observers, altering callback tables or driver hooks, so system-wide visibility disappears without triggering alarms.
- **EDR INTERNALS:** Tamper with the telemetry pipeline or in-memory reports, so the agent thinks everything is fine. It's like turning on the red "recording" button, but nothing is being recorded.

WFP Callout Removal:

Windows Filtering Platform (WFP) removal is similar to yanking the security checkpoint out of an airport while leaving the metal detectors standing. Traffic still flows; everything looks operational, but nothing is actually being screened or flagged.

- **USER LEVEL:** Nudge the user-space components that handle network telemetry, no alerts, no logs, just smooth and suspiciously quiet connections.
- **KERNEL LEVEL:** Remove or patch the WFP callouts that inspect packets and enforce filtering policies. The network stack keeps humming, but the inspectors are off duty.
- **EDR INTERNALS:** Cut the link between the WFP engine and the agent's network visibility modules, so it never hears about blocked or suspicious traffic.

EDR internals tamper with the telemetry pipeline or in-memory reports, so the agent thinks everything is fine. It's like turning on the red "recording" button, but nothing is being recorded.

Looping Sabotage of a Hot Path:
The equivalent of forcing the security conveyor to slide the same suitcase past the X-ray a dozen times while new bags pile up. The critical processing lane gets stuck in a loop, so fresh telemetry either isn't processed or is delayed until it's useless.

- **USER LEVEL:** Inject or nudge user-space logic (event loops, queues, or agent threads) so they repeatedly reprocess the same item, sleep, or spin effectively, throttling or starving normal event forwarding.
- **KERNEL LEVEL:** Create spin-loops, repeated callbacks, or lock contention in kernel callbacks or driver hooks that occupy the hot path. CPU/time slices get wasted on rework, so kernel-level visibility and timely handling evaporate.
- **EDR INTERNALS:** Corrupt or bloat the telemetry pipeline or buffers, so events are retried, deduplicated, or sampled by the agent, which delays, aggregates, or silently drops alerts because the processing pipeline is stuck replaying old work.

Stopping these attacks is mostly about keeping the kingdom locked and double-checking the guards. Signed drivers and patched kernels make it harder to sneak in rogue code, while monitoring mini-filters and WFP callouts ensures the cameras are actually recording. Hot-path loops get tamed with watchdogs and timeouts, and cross-agent telemetry catches anything trying to slip through. Layered defenses, tamper protection, and behavioral checks mean that even if one alarm gets unplugged, the system still notices something is not right.

Stopping these attacks is mostly about keeping the kingdom locked and double-checking the guards.

3 LOTL: Weaponized Trust to Blend In

Sadly, ransomware groups that like Living off the Land aren't looking to homestead in Alaska. Living off the Land (LOTL) tactics use what's already built into your OS against you. Threat actors who have already established initial access and privilege escalation will then leverage tools such as:

- PowerShell
- WMIC
- Rundll32
- PsExec

Ransomware groups rely on legitimate system tools to execute malicious actions. Using WMI, macros, or admin tools like PsExec, they move laterally, persist, and execute payloads without dropping obvious malware. Like a burglar using the building's doors, elevators, and even 'acquired' janitor keys, endpoint systems see activity, but nothing looks out of the ordinary. Fileless execution, reflective DLL injection, and legitimate network channels let attackers operate in plain sight, leaving SOC teams relying on behavioral monitoring and anomaly detection.

Defending against LOTL attacks is basically about childproofing your own toolbox. Lock down what's already in the house, let approved scripts and binaries run, and don't hand out PowerShell and WMI access like candy. Strip unnecessary admin rights, log everything, and read those logs when Word suddenly spawns a PowerShell process at 2 a.m. Correlate endpoint, network, and identity data so "perfectly normal" activity starts looking suspicious when it forms a pattern. In short: don't let attackers borrow your own tools to rob you. Keep them locked up, labeled, and under surveillance.

4 Process Injection: No Files - No Problem

Process injection and fileless attacks are like ghosts; they slip their code into trusted programs or run it straight from memory, leaving no files behind and making your system think everything's perfectly normal. Often, they load and run malicious code directly in memory or sew it into trusted processes using tricks like:

- Reflective DLL loading
- Process hollowing
- DLL injection
- Remote threads/APC queuing
- Tiny in-memory loaders kicked off by a single macro or PowerShell one-liner

Living in memory lets the attacker bypass disk-based antivirus and endpoint scanners, leaves few forensic artifacts for investigators to find, and lets the malicious activity look like normal system behavior by reusing built-in execution hosts (PowerShell, WMI, msbuild) and faking benign parent-child chains. Common signals to keep an eye out for are:

- Mismatches between on-disk file hashes and the in-memory image of the same process.
- Unexpected modules or nonstandard code loaded into trusted processes (svchost, explorer, etc.).
- Many short-lived parent/child processes or strange, compact command lines that look like one-shot loaders.
- Unusual memory allocation patterns (large RWX allocations, repeated virtual allocations) and atypical thread creation (remote threads, APC injections).

This type of attack is the equivalent of threat actors slipping in by hiding trusted guests already allowed in the house. To address this, lock the doors, only let verified apps like signed software run, turn on alarms that watch memory for hidden payloads, and disable old chatty tools like PowerShell v2. Also, keep an eye on any strange parent-child app behavior, such as Notepad suddenly controlling system processes. In short, trust but verify, watch insiders, and this will help bring invisible attacks out in the open.

5 Evidence Destruction: Leave No Trace Behind

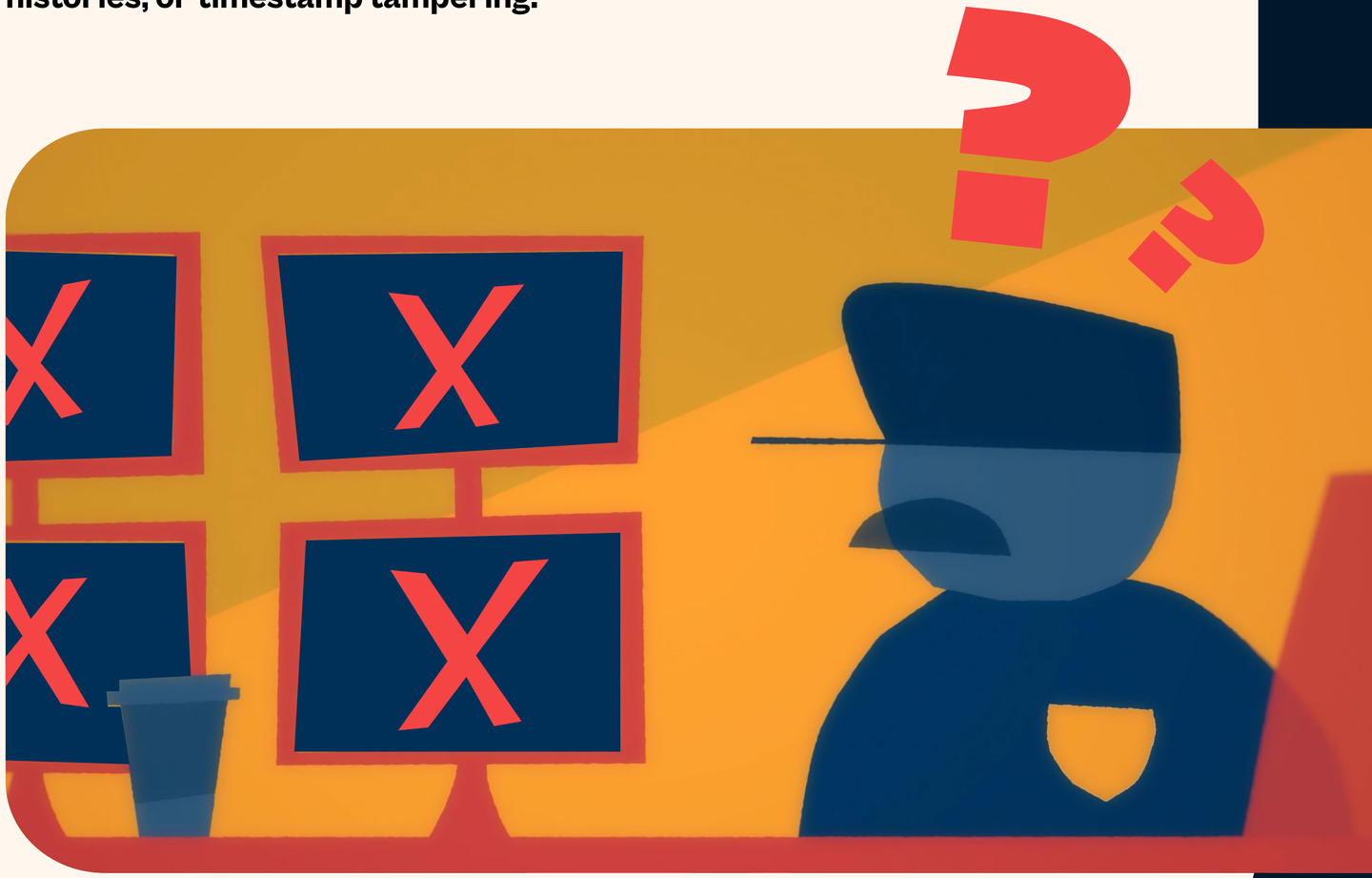
Threat actors blind EDR sensors and destroy evidence in hopes of turning your EDR into a misguided witness. Methods used can include:

- Unhooking or disabling SIEM API monitors
- Disabling logging services
- Delete and truncate logs
- Deleting snapshots and backups
- Blocking agent to console communications

Evidence destruction commonly involves overwriting event logs, clearing Sysmon histories, or timestamp tampering. They erase the forensic trail so completely that investigators are left starring in CSI: The Episode with No Evidence. The goal here is to buy time and delay detection during initial access and lateral movement, giving attackers ample time to dwell undetected, exfiltrate data, and deploy ransomware before anyone takes notice.

So, what is the best defensive strategy here? Ship logs to immutable storage the second they're generated, avoiding local tampering. Shield backup credentials with ironclad access controls and stash copies in air-gapped, immutable vaults. Set alerts for log gaps or forwarder hiccups; the name of the game here is flagging any period of silence. Finally, lock any change to logging or backups behind strict change control, avoid the mysterious maintenance without a ticket trail and plenty of eyes watching.

Evidence destruction commonly involves overwriting event logs, clearing Sysmon histories, or timestamp tampering.



Defensive Maneuvers for Evasive Tactics

EDRs stand guard on the frontlines of endpoint defense. But ransomware groups aren't coming in through the front door; they're rewriting the rules to slip past unnoticed. Halcyon changes the equation. Built to detect and defeat ransomware even when traditional defenses are blinded or bypassed, Halcyon provides resilience where EDRs reach their limits. By focusing on ransomware's unique behaviors and recovery pathways, Halcyon ensures that when attackers rewrite the rules, defenders still come out on top.

Ready to see how resilience really looks in action?
Book a Halcyon demo and find out.

halcyon