# Advantages of Agentless Linux EDR

Sandfly Security®

# Advantages of Agentless Linux EDR

2026-01-07

## Introduction

Linux Endpoint Detection and Response (EDR) is dominated by a kernel-level agent-based security model inherited from the Windows world. While agent-based security has certain advantages, it also carries significant risks on Linux that often means it cannot be deployed in many mission critical and mixed environments. Sandfly addresses these risks through the use of innovative agentless EDR on Linux. By not using an agent, Sandfly is able to get excellent reliability, minimize compatibility risks, and cover more Linux systems than any agent-based approach on the market.

This paper discusses the ten most critical advantages of the agentless model, illustrating why it is a stable, performant, and cost-effective choice for comprehensive Linux security vs. using traditional endpoint agents.

## Top 10 Advantages of Agentless EDR on Linux

1. **Low Risk of Kernel Panics**
   The agentless approach runs entirely in user space, meaning it never loads kernel modules, hooks system calls, relies on eBPF, or interfaces with the operating system's core directly. This architecture makes it extremely unlikely for agentless EDR to cause a server crash (also called a kernel panic), which is a well-documented risk with kernel-level agent-based EDR on Linux. If an agentless scan fails, it simply logs an error and stops. It cannot enter a persistent loop that crashes a system on boot or cause other failures. This is a fail-safe error handling which is critical for high-availability production servers where downtime is unacceptable. Agentless stability risks are negligible, contrasting sharply with the risks introduced by a persistent agent in the Linux kernel space.

2. **Universal Compatibility Across Linux Architectures**
   The agentless model connects via standard SSH and is designed to work on virtually any Linux system, regardless of its age or architecture. This includes legacy servers (e.g., 10+ years old), embedded devices, IoT, and systems running non-x86 CPUs like ARM or MIPS. Of course Sandfly's agentless model also supports modern cloud and on-prem based

servers. Agent-based EDRs suffer from limited compatibility due to strict kernel version requirements, often failing on mixed distributions, older kernels that cannot be upgraded, or custom environments. Agentless solutions can secure many more Linux systems vs. traditional agents, eliminating critical blind spots that exist in many enterprises today.

3. **Minimal Performance Overhead**
   Agentless EDR systems have zero CPU and RAM overhead when they are not actively scanning a host. When a scan is running, it is a tiny, non-persistent burst, often less than 1-2% aggregate CPU time running with low-priority which lasts around 60-120 seconds before disappearing. Agent-based solutions are always on, requiring continuous use of system resources (we've seen ranges from 5% to over 30% CPU). This persistent resource consumption adds up in high capacity environments, translating into significant, wasted compute cost and performance impacts that can impact mission-critical operations.

4. **Kernel Update Conflict Risk**
   Because Sandfly's agentless EDR does not touch the kernel, it works immediately on *any* new kernel version the day it is released, and it never interferes with OS patching. Agent-based tools are susceptible to "Kernel Hell," where new kernel releases often break the EDR agent's compatibility, forcing the agent vendor to issue an update. This means organizations enter a paradox where they must sometimes hold back OS security patches just to keep the EDR agent functioning, creating a significant security liability.

   Further, Linux has had many **thousands of kernel releases** since it was introduced. Even worse, each distribution will build its own kernel from sources. Kernels that appear to have the same version number may in fact have different patches, drivers, and other changes applied by the vendor that can cause agent compatibility issues.

   With this in mind, it is simply impossible for an agent-based EDR vendor to test against all the permutations, architectures, patch-levels, and more to ensure absolute stability when doing kernel monitoring. This results in a much narrower Linux distribution support than an agentless solution can provide. Agentless solutions are not bothered by kernel updates and have extremely low risk of any system stability impacts regardless of update schedule from Linux vendors.

5. **Instant, Zero-Friction Deployment**
   Deployment of an agentless EDR is instant and remarkably simple—it only requires providing SSH credentials to the system you want protected. Customers can onboard thousands of servers in minutes without installing a single package, rebooting any system, or performing any local configuration.

Agent-based deployment, in contrast, is complex, risky, and time-consuming, often taking weeks or months of planning, automation scripting, package management, and dependency troubleshooting for an entire enterprise fleet. Each endpoint must be touched to install agent-based systems and this carries significant risk in an enterprise. Even worse, each endpoint must be touched again whenever the kernel or agent gets an update making this risk a continuous threat to enterprise stability.

Due to the rapid nature of agentless deployment, it also makes it uniquely valuable for incident response when immediate visibility into compromised systems is necessary. This is especially true as loading an agent during a live incident just adds additional stress to an already stressful situation and often cannot or will not be done by security and operations teams.

6. **Evasion Resistance**
   Agent-based solutions are a permanent, known target that are increasingly targeted by attackers with blinding attacks to outright uninstallation of the security protection itself. Agentless EDR functions as an independent, external forensic investigator which shows up on hosts by surprise or on-demand without dependencies on the host. Sandfy's agentless protection treats the operating system (OS) as untrusted. It executes active threat hunting checks looking for discrepancies between what the OS reports and what the underlying systems show. This external view allows it to "decloak" sophisticated Linux malware, such as kernel-level rootkits, that are designed specifically to blind, disable, or lie to persistent EDR agents running inside the compromised kernel. By avoiding installation of an agent directly, systems also appear unprotected to attackers which can lower their guard and allows agentless systems to avoid evasion tactics taking intruders by surprise.

7. **Superior Detection of Stealthy, Post-Compromise Threats**
   Sandfly's agentless solution is tactics detection-focused, using methods written by Linux experts to find post-compromise tactics like kernel rootkits, persistence mechanisms (e.g., cron/systemd abuse), credential theft, unauthorized binaries, and more. This tactics-hunting style generates low false positives because it is natively focused on Linux attack patterns without any Windows baggage. Many agent-based EDRs are multi-platform tools that are Windows-centric, and their detection models can be superficial or prone to generating high noise levels on legitimate Linux admin activity, making it harder to spot real threats.

8. **Incident Response and Integrity Validation**
   Sandfly's agentless approach acts as an indispensable "second opinion" or independent auditor for an organization's existing security tools. Since Sandfly operates externally and verifies the reality of the system, it can detect if a primary agent-based EDR has been blinded, tampered with, or silently disabled by a sophisticated rootkit or other mechanisms. It can also ensure the system has not drifted from a known-good configuration. This integrity validation ensures that the organization's existing security investments are actually working and that the primary EDR dashboard's "green/healthy" status is not a dangerous lie being fed by a compromised host.

   Further to this, Sandfly can find what we call "dormant" attacks which are changes done to a Linux host that are malicious but are not active in memory and therefore would be missed by traditional EDR solutions. We can detect suspicious dropped files, compromised SSH keys, weak passwords, suspicious users, malicious scheduled tasks, and more. This is especially useful for incident response where a system may have been compromised in the past but there was no security monitoring in place.

9. **Seamless Support of Varied Linux Architectures**
   Agentless EDR supports modern and varied Linux environments, including embedded systems, networking gear from makers like Juniper or Cisco, cloud-native environments, and container hosts. It works perfectly where SSH exists, providing full visibility without special configuration. Agent-based vendors struggle with the fragmentation and unpredictability of the Linux world, often requiring painful workarounds or simply no coverage which leaves gaps in security.

10. **Dramatically Lower Total Cost of Ownership (TCO)**
    While agent-based tools quote a per-host licensing fee, they hide a massive operational tax that dramatically increases the total cost of ownership. This hidden cost includes the time spent on agent management teams, continuous compatibility testing against new kernels, performance tuning to mitigate resource theft, and costly rollback drills. The agentless model removes this burden completely, requiring no local endpoint maintenance, resulting in a significantly lower TCO in real enterprises, making it a more cost-effective and faster solution for Linux coverage.

These points summarize the main problems with agent-based solutions which we've seen both directly and through customer experiences. When agent-based solutions work, they can be effective, but often there are serious and persistent problems with agents on Linux. Simply not having an agent eliminates many serious risks and pitfalls for security visibility on this platform. The comparison chart below summarizes these findings.

Sandfly Security®

# Comparison Chart

| Feature | Sandfly Security | Agents |
|---|---|---|
| **Architecture** | **Easy:** Ephemeral binary pushed via SSH. Executes and then removes itself. | **Complex:** Heavy, persistent kernel-level agent that requires installation and maintenance on each system. |
| **Stability Risk** | **Low:** Runs in user-space. Extremely low stability or performance risks. Proven on critical infrastructure globally. | **High:** Kernel panic risk is real and documented. Performance impacts are common and can be severe. |
| **Compatibility** | **High:** Legacy, IoT, Cloud, On-Prem, AMD, Intel, MIPS, ARM support, etc. Can run on just about any Linux based system. | **Low:** Specific kernels & distros (mostly RedHat and a few others). Won't work on custom Linux versions, embedded systems, etc. |
| **Performance Impacts** | **Low:** Sandfly has been extensively tested in latency critical applications and shown to have little to no discernable performance impacts. | **High:** Monitoring kernel level activity is always going to impact performance. Latency and other performance impacts are very common and sometimes severe, which can disrupt operations. |
| **Kernel Conflicts** | **Low:** Does not tie into the kernel and is not affected by kernel updates or customizations. | **High:** Kernel versions and updates frequently require extensive compatibility testing. Updates can break agents, or agents can break kernels. |
| **Deployment** | **Instant:** No installation required. Point it at systems and go. | **Risky:** Requires package install & ongoing testing for each update. Often requires reboots and updates require touching each endpoint again and again. |

| | | |
|---|---|---|
| **Evasion Resistance** | **High:** Non-persistent presence on host makes it harder to target for disabling or blinding. | **Low:** Increasingly targeted by sophisticated attackers with blinding to disable telemetry. |
| **Detection Style** | **Tactics Hunting:** Hunts compromise tactics (rootkits, persistence, lateral movement risks). Designed from the ground up for Linux attack detection. | **Malware Signatures:** Often uses Windows-centric traditional malware signature thinking which is weak against constantly changing open source Linux tools and tactics. |
| **Incident Response** | **Fast:** Can be deployed during live incidents to investigate hosts without introducing new risks to an already stressful situation. | **Slow:** Need to load agents on systems potentially impacted by attackers. This can destroy evidence or cause stability problems delaying response. |
| **Monitors Network Appliances and IoT** | **Yes:** Sandfly supports most CPU architectures of Linux which means it can easily watch a server, network switches, or even a low-powered IoT device like cameras. If it runs Linux and has SSH, Sandfly can likely run on it. | **No:** Agents need extensive testing and rarely run on non-server hardware. Installing agents can void vendor warranties and support contracts. |
| **Total Cost of Ownership (TCO)** | **Low:** With no kernel dependencies, reliability testing is easy and fast. No need to touch or maintain agents on each endpoint. | **High:** Each new kernel or agent update needs comprehensive testing to ensure compatibility and ongoing maintenance. |

# Conclusion

In summary, Sandfly's agentless approach not only eliminates the risks and operational drag associated with kernel-level agents but also delivers a level of stability, compatibility, and evasion resistance that is essential for modern, mission-critical Linux defense. For any organization prioritizing stability and comprehensive coverage over the inherent risks of agent deployment, agentless EDR is the definitive path forward.