EvilEDR: Repurposing EDR as an Offensive Tool

Alachkar, Kotaiba; Gaastra, Dirk; Barbaro, Eduardo; van Eeten, Michel; Zhauniarovich, Yury

DOI
10.5555/3766078.3766109

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the 34th USENIX Security Symposium

USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

EvilEDR: Repurposing EDR as an Offensive Tool

Kotaiba Alachkar, *Delft University of Technology;* Dirk Gaastra, *Independent Researcher;* Eduardo Barbaro, Michel van Eeten, and Yury Zhauniarovich, *Delft University of Technology*

https://www.usenix.org/conference/usenixsecurity25/presentation/alachkar

This paper is included in the Proceedings of the
34th USENIX Security Symposium.

August 13–15, 2025 • Seattle, WA, USA

978-1-939133-52-6

Open access to the Proceedings of the
34th USENIX Security Symposium is sponsored by USENIX.

ARTIFACT
EVALUATED
usenix
ASSOCIATION
AVAILABLE

EvilEDR: Repurposing EDR as an Offensive Tool

Kotaiba Alachkar
Delft University of Technology

Dirk Gaastra
Independent Researcher

Eduardo Barbaro
Delft University of Technology

Michel van Eeten
Delft University of Technology

Yury Zhauniarovich
Delft University of Technology

Abstract

Endpoint Detection and Response (EDR) systems provide continuous monitoring, threat detection, and response capabilities. This has driven their widespread adoption in enterprises, making them a key part of an enterprise's security architecture. However, EDR systems are a double-edged sword, and in this study, we demonstrate how this class of systems can be employed for offensive use. Unlike prior studies that focused on evasion and tampering, we introduce the new concept of *EDR repurposing*, which we call *EvilEDR*. Our analysis shows that EvilEDR can be used to execute arbitrary commands via the response console, transfer tools, exfiltrate data, and passively collect system information to facilitate further exploitation and lateral movement. EvilEDR operates covertly, masquerading as a legitimate process and communicating seamlessly with trusted domains. Additionally, we show that EvilEDR can impair defenses by registering its own EPP as the default. It can also isolate the host from the network, severing telemetry and response channels essential for enterprise defense mechanisms. Fortunately, EvilEDR can be effectively detected and mitigated, and in this paper, we propose concrete and actionable defense strategies to achieve this.

1 Introduction

For decades, AntiVirus (AV) software was the primary defense for endpoint security, relying on signature-based and heuristic detection to block known malware. As threat actors adopted more advanced and evasive techniques [33, 49], traditional AV struggled to keep up. This gap, particularly in enterprise environments with large numbers of endpoints and complex networks, led to the development of Endpoint Detection and Response (EDR) solutions. EDR provides real-time monitoring, behavioral analysis, and rapid threat response across multiple endpoints.

In this work, we introduce *EDR repurposing*, a novel approach that utilizes an EDR system as an offensive tool instead of its original defensive purposes. Unlike EDR evasion and tampering [2, 77], which aim to disable EDR protection, EDR repurposing uses the legitimate capabilities of the EDR itself for malicious purposes. Using an off-the-shelf attacker-controlled EDR, which we call *EvilEDR*, attackers can carry out malicious activities while remaining undetected. It is important to note that EvilEDR does not exploit any flaw or vulnerability in the EDR software; instead, it uses its inherent features for malicious purposes. This technique highlights a paradigm shift in offensive security, driven by the concept that repurposing EDR is far more efficient and effective than bypassing or disabling it.

While repurposing benign tools is not a novel concept, the use of EDR is new and noteworthy. First, its trusted status as a security tool facilitates the distribution and detection of evasion by operating alongside an enterprise-controlled EDR. Second, its deep integration into the OS and rich capabilities – particularly extensive system activity monitoring and built-in remote access through a live response console – make it a powerful Swiss army knife for attackers, serving as an attractive substitute for widely used offensive tools and frameworks. Third, it has increased persistence compared to traditional offensive tools, inherently designed to maintain a strong foothold for defensive purposes. This combination of properties makes EvilEDR an exceptional case that warrants increased awareness.

In this study, we analyze four EDR solutions: Microsoft Defender for Endpoint (MDE), Elastic Defend, Sophos EDR, and Trend Micro Apex One. We show that each of them could be employed for offensive purposes. We demonstrate that EvilEDR can be used to maintain access to the target system, enabling command and control operations, malicious code execution, data manipulation (e.g., ransomware), and exfiltration of sensitive information such as credentials and critical system files. We identify a novel attack path, *host isolation*, where EvilEDR severs the Enterprise EDR's telemetry collection to weaken its defense capabilities. Also, EvilEDR can override existing Endpoint Protection Platform (EPP) protection by registering its own EPP, deliberately weakened by the attacker, to obtain greater control over the system.

We evaluate EvilEDR's effectiveness through a case study resembling real-world attack scenarios. We show this on a testbed replicating a typical *secure enterprise environment* with both EPP and EDR systems. Each EDR solution was used interchangeably as Enterprise EDR and EvilEDR to ensure broader applicability. Our experiments show that each EvilEDR operates undetected alongside the Enterprise EDR solutions, demonstrating its significant impact in practical scenarios.

Furthermore, we propose a multi-layered approach for enterprises to prevent and detect EvilEDR. Preventive measures aim to hinder EvilEDR deployment and execution, while detective measures focus on identifying its drivers and processes. We provide practical implementations of these detection measures and evaluate their effectiveness across the four EDR solutions covered in this study. For vendors, we propose measures to limit the misuse of their EDR software.

In short, we make the following contributions:

- To the best of our knowledge, this paper presents the first study on using EDR systems for offensive purposes. We perform a detailed analysis of how EDR capabilities can be exploited.
- We evaluate the EvilEDR's effectiveness in real-world attack scenarios within a secure enterprise environment.
- We propose defense strategies for both the enterprise and vendor levels. Additionally, we provide artifacts that can be used to implement detection measures.

Our primary goal is to raise awareness of this novel attack technique and help enterprises and EDR vendors recognize and defend against the threat posed by EvilEDR. We focus on common EDR features that can be used for EvilEDR, rather than a comprehensive evaluation of vendor-specific features. Although we provide an evaluation matrix for the four EDR vendors covered in this research, we focus on how these features can be misused for EvilEDR.

2 Background

The rise of polymorphic malware and packed or encrypted malicious files rendered traditional AV solutions ineffective as the sole endpoint defense. This led to the development of Next-Generation AV (NGAV), which expanded signature-based methods with behavioral and cloud-based analysis for malware detection. As attack techniques evolved into multi-stage steps [33,49], a single-point security approach proved inadequate. To improve defense, enterprises adopted EPPs [20], which integrate NGAV with data encryption, host-based firewalls, intrusion prevention, device control, and more into a unified, centrally managed system. However, EPP focuses on preventing initial infections and lacks advanced threat detection and response capabilities. As endpoint threats grew more sophisticated [55] and enterprise environments more complex, EDR emerged to complement EPP and became an integral part of enterprise security architecture [5, 32]. EDR works alongside EPP and provides real-time visibility, advanced threat detection, and response across endpoints. To remain effective, EDR vendors continuously make functional advancements to improve their solutions' capabilities, features, and detection coverage[1].

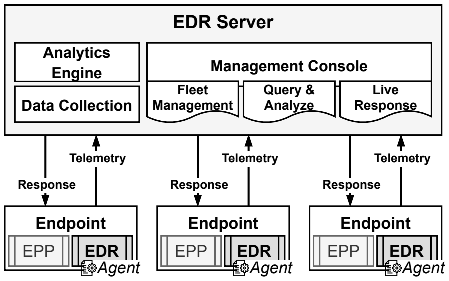

Figure 1: EDR System Architecture Overview

System Architecture. Figure 1 presents the architecture of an EDR solution. EDR systems roughly consist of two key components, *software agents* installed on endpoints and a *server*, which can be either self-hosted or cloud-hosted in the vendor's environment. The software agents operate in a dual-mode architecture. Locally, agents collect endpoint telemetry and apply built-in rules and heuristics for threat identification. In addition, they send telemetry to the EDR server for further analysis. The EDR server collects this telemetry and applies a variety of advanced analytics and Machine learning (ML) techniques to identify suspicious behaviors and potential threats. This setup enables real-time on-device threat detection and prevention while offloading more intensive processing to the server. The EDR server also provides fleet management to centrally manage endpoints and includes query and analysis tools to search and monitor events and endpoint activities. When a threat is detected, the EDR system generates alerts, initiates automated responses, and provides tools for the security team to manually investigate and remediate when necessary. Additionally, EDR enables security teams to proactively hunt for threats that may not have been automatically detected through in-depth investigation of endpoint activities.

Functional Capabilities. EDR capabilities vary across vendors, but most share the same core features. First, they integrate deeply with the Operating System (OS) through kernel-level drivers, providing in-depth visibility into system activities and behaviors. This integration also enables tamper protection and self-defense mechanisms, ensuring that the EDR remains operational even during an active system compromise. Second, EDR solutions provide both automated and manual response capabilities. This is done by intervening in

[1] https://learn.microsoft.com/en-us/defender-endpoint/whats-new-in-microsoft-defender-endpoint

malicious behavior directly, or providing an enterprise's security team with the right tools to investigate and respond to a potential threat. These features allow EDR to maintain visibility and protection even during the later stages of an attack. For example, if an attacker gains initial access through social engineering or by compromising a valid account, EDR's tamper protection prevents them from disabling its services, even if they escalate privileges or obtain administrative access. Instead, the EDR remains active, continuously monitoring activities and hindering the attacker's ability to establish persistence or move laterally within the network.

OS-Level Access. EDR agents often run with the highest privileges [75] within the OS, operating in both kernel and user mode to monitor and intercept threats before they impact the system [78]. In kernel mode, EDR uses a driver loaded into the Windows kernel to monitor system calls, process creation, network connections, registry changes, file activities, and memory access. It also uses Event Tracing for Windows (ETW) to receive real-time notifications of system events, providing in-depth visibility of system activities and efficient, low-overhead monitoring.

Trusted Status. EDR systems are inherently trusted tools, a status reinforced by their role as security solutions and the reputation of well-established vendors. On a technical level, EDR systems gain their trusted status through deep OS integration, achieved by passing rigorous certification processes to meet kernel-level access requirements, and digital signing of their software and processes. Notably, to obtain kernel access, EDR vendors undergo rigorous certification processes to ensure their drivers meet security standards and comply with Microsoft's requirements[2]. This approach prevents malicious actors from gaining kernel access through drivers and limits kernel access exclusively to a select group of vendors [51]. These measures ensure authenticity, tamper resistance, and privileged operation on endpoints. Combined with their widespread adoption, these characteristics establish EDR systems as a deeply integrated and reliable component within enterprise environments.

2.1 Related Work

Our research presents a novel approach to repurposing EDR systems as attack tools, a technique that, to our knowledge, has not been directly explored before. The closest work, which we have identified, is from SafeBreach Labs [75], who explored how to bypass anti-tampering mechanisms in Palo Alto Networks Cortex EDR and demonstrated how EDR behavior can be exploited to persist malware. However, their focus was on exploiting software vulnerabilities. In contrast, our approach leverages the inherent trust and built-in capabilities of EDR systems, making it broadly applicable across various EDR solutions without depending on software flaws. We reviewed related work in EDR evasion and tampering, Living-Off-The-Land (LotL) techniques, and the malicious use of legitimate tools, distinguishing between academic research and open-domain studies due to their varying scopes.

For EDR evasion and tampering, academic research [8, 17, 29, 35, 41, 44, 46] primarily evaluates EDR evasion techniques. Open-domain research, conducted by independent security researchers, focuses on practical demonstrations of EDR evasion [9, 23, 24, 42, 58, 59, 65, 72, 76, 77, 79, 86, 89] and tampering [4, 25, 38, 48] techniques. While these studies focus on evasion and tampering, our research explores repurposing EDR's inherent capabilities for offensive use. Note that while security providers continually enhance their products to counter evasion and tampering [13], they cannot effectively prevent EDR systems from being repurposed without knowing a particular context in which their tools are deployed and the accounts they are associated with.

For LotL, academic research [7, 12, 27, 67, 69, 83, 88] has primarily focused on evaluating LotL attack techniques and exploring various detection and mitigation approaches. Open-domain research provides practical and up-to-date LotL techniques and tools, such as the LOLBAS Project[3] for Windows binaries, scripts, and libraries, the LOLDrivers Project[4] for Windows drivers, and the GTFOBins Project[5] for Linux. While effective detection and mitigation strategies for using LotL techniques have been developed [45], our research distinguishes itself by extending beyond the traditional LotL focus to explore the repurposing of EDR systems. Unlike LotL, which relies on pre-existing tools within the operating system, EDR repurposing involves installing additional tools that are not native to the system, thereby offering broader capabilities and applications for offensive operations.

Lastly, academic research [18] has shown how legitimate tools can be weaponized for ransomware attacks. Reports [21, 22] from the Cybersecurity and Infrastructure Security Agency (CISA) highlight the misuse of legitimate software, particularly Remote Monitoring and Management (RMM) tools in attacks. Open-domain research further details attack techniques and Advanced Persistent Threat (APT) campaigns exploiting legitimate software and dual-use tools [1, 6, 43, 47, 68, 73, 80, 84]. Although detection and mitigation strategies for the misuse of legitimate applications have advanced [10, 37], our research takes a different approach by repurposing EDR capabilities and leveraging its trusted status. This novel technique, not previously explored, effectively demonstrates its potential across multiple attack stages.

[2] https://learn.microsoft.com/en-us/windows-hardware/drivers/install/driver-signing

[3] https://lolbas-project.github.io/

[4] https://www.loldrivers.io/

[5] https://gtfobins.github.io/

3 EvilEDR

As we discussed in Section 2, in order to respond to malicious threats, modern EDR systems are equipped with rich functionalities: EDR agents can launch processes, run custom commands, download and upload files according to the directions issued by the EDR server. Given such a wide range of capabilities, which can be exploited for malicious purposes, the idea of EvilEDR becomes obvious: *deploy the EvilEDR agents on endpoints and control them through your own EvilEDR server, which acts as a central hub for managing compromised endpoints*. The concept of EDR repurposing is *simple*, *effective*, and *highly applicable*.

Simple. Repurposing is simple as it does not rely on exploiting software vulnerabilities or flaws. It also eliminates the need to develop custom tools and extensions or maintain complex infrastructure. Instead, attackers can access off-the-shelf EDR solutions, deploy them on target endpoints, and misuse their features for offensive purposes. For instance, the live response console, typically used for manual investigations, can be repurposed into a remote command execution tool, functioning similarly to a Command and Control (C2) framework.

Effective. The repurposing approach is effective because it leverages the trusted status of EDR, its in-depth OS access, and its rich capabilities. For instance, the EvilEDR agent binaries are code-signed with a valid and trusted certificate, therefore their download and installation process does not trigger alerts from existing security tools. OS-level access provides attackers with full visibility into system activities without relying on techniques or tools that could trigger alerts. Furthermore, EDR's built-in features can replace common attack frameworks; for instance, the file download functionality can be used for data exfiltration, eliminating the need for external file transfer tools that could trigger alerts. Most importantly, EvilEDR can operate alongside Enterprise EDR, allowing attackers to perform malicious actions under the guise of legitimate software, thereby remaining undetected.

Applicable. EDR repurposing remains highly relevant even when attackers obtain administrative privileges on a system. While privileged access provides significant control, Enterprise EDR solutions remain active, monitoring system activities and hindering the attacker's progress. Attempting to disable or bypass Enterprise EDR is both complex and likely to trigger alerts that could expose the attack. Similarly, deploying other malicious tools is likely to be detected and flagged by Enterprise EDR, further increasing the risk of exposure. EvilEDR offers a stealthier alternative, optimizing the attack lifecycle while reducing detection risks, particularly during persistence, lateral movement, command and control, and data exfiltration stages.

3.1 EvilEDR Setup

The specific setup of EvilEDR depends on the chosen EDR solution. A key requirement is that EvilEDR and Enterprise EDR must be different solutions, as two instances of the same EDR cannot coexist on a single machine. Access to EDR solutions can be obtained through trial licenses, paid subscriptions, or free/open-source options.

For the server environment, some vendors, such as Elastic, offer self-hosted options that attackers can deploy on a virtual machine (VM). Others, like MDE, are only provided as a cloud (SaaS) solution. The choice between self-hosted and cloud-hosted depends on the desired stealthiness level and flexibility. With self-hosted servers, the EvilEDR agent communicates with attacker-controlled IPs or domains, making detection easier. In contrast, cloud-hosted EvilEDR traffic appears to communicate with legitimate vendor-owned domains, such as those managed by Microsoft, making it harder to detect. This distinction is particularly critical in environments where network traffic is actively monitored. At the same time, cloud-based solutions are easier for vendors to disable or revoke access if EvilEDR is identified in an active attack campaign and reported to them.

EvilEDR configuration requires no specific setup. To enhance stealthiness and effectiveness, it is recommended to switch security policies from block mode to audit mode, or even disable them entirely. This helps prevent any unexpected behavior that might alert users to EvilEDR's actions. Additionally, disabling cloud and telemetry submissions to the EDR vendor, when possible, minimizes the risk of the vendor detecting the attacker's activities. These configurations vary by vendor, but from an attacker's perspective, EvilEDR settings should be intentionally weakened to maximize flexibility and reduce the chances of detection or unexpected behavior.

As for the EvilEDR agent installation, attackers with admin privileges can directly install the EvilEDR agent using the vendor's standard installer. Alternatively, they can trick users into installing it through phishing or social engineering. Some EDR vendors [16, 92] offer a feature to invite users via email to download and install the EDR agent, which can be exploited to facilitate phishing or social engineering attacks.

3.2 Testbed

Our testbed, shown in Figure 2, simulates a secure enterprise environment, consisting of Windows 11 endpoints running the latest version (22H2) and a domain controller running Windows Server 2022 (used in the case study in Section 5). All Windows 11 endpoints were protected by a Windows Defender EPP running in block mode and an Enterprise EDR. We used the same four EDR solutions – MDE, Elastic, Sophos, and Trend Micro – as Enterprise EDRs. We selected these EDR solutions because they participated in the MITRE ATT&CK Turla Enterprise Evaluation 2023 [60], discussed

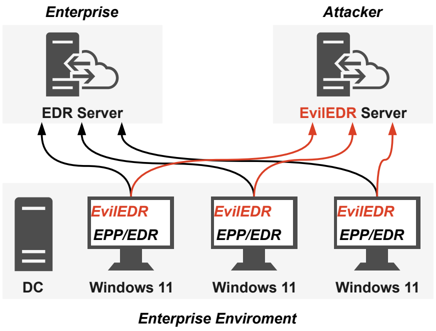

Figure 2: Testbed Overview

further in Section 5. All pairwise combinations of these solutions were evaluated, with one acting as Enterprise EDR and the other as EvilEDR. When an EDR solution was used as Enterprise EDR, tamper protection was enabled, and all security features were set to prevention mode. When configured as EvilEDR, all cloud submit features were disabled, and security features were either turned off or set to audit mode.

To demonstrate EvilEDR potential, we compared its stealthiness and effectiveness against actions performed via an administrative command prompt. To ensure a clean and consistent environment during our tests, we used Ludus[6] to manage our cyber range. We provide configuration details as artifacts.

3.3 Threat Model

We assume a threat actor targets a secure enterprise environment with endpoints running up-to-date software, protected by both EPP and EDR solutions, with all security features, tamper protection, and strict detection and prevention rules enabled. We assume these endpoints are used by end-users and therefore, do not have strict network access restrictions such as URL allowlisting, nor do they have overly restrictive software installation policies applied. We assume the attacker has gained initial access to the endpoint through valid accounts [61], phishing [62], exploiting vulnerabilities[7], or an insider threat [70], and has elevated privileges to install software locally.

The attacker's objectives include establishing C2, data exfiltration, and lateral movement. We do not assume the attacker's knowledge level, as it could range from a low-skilled actor with basic tools to a nation-state adversary with access to advanced techniques and insider information. We assume the attacker has no direct access to the Enterprise EDR system itself but has limited yet realistic knowledge of the enterprise's EDR system and network configurations, acquired through reconnaissance and system exploration. Lastly, the enterprise maintains active monitoring and incident response, which the attacker aims to evade.

4 EvilEDR Attack Potential

In this section, we use the MITRE ATT&CK Matrix for Enterprise [64] to demonstrate the attack potential of EvilEDR. While we primarily focus on EvilEDR's operation post-initial access and privilege escalation, we also highlight its role in facilitating earlier attack stages, especially when existing security defenses are in place. Table 1 provides a summary of EvilEDR's functionalities supported by the tested EDR solutions. It is important to note that the techniques covered are not exhaustive; instead, we focused on the most significant ones to demonstrate EvilEDR's attack potential.

4.1 Initial Access to Privilege Escalation

As already pointed out in Section 3.1, attackers can either install the EvilEDR agent directly if they already have elevated privileges or trick users into installing it. In the latter case, EvilEDR proves being useful during *initial access* attempts involving phishing due to its trusted status as a security tool. This trust, combined with the valid signing of its agent software, helps bypass enterprise prevention controls during download and installation, reducing the risk of detection. Once deployed, EvilEDR can also assist attackers in the *execution*, *persistence*, and *privilege escalation* stages.

Execution. EvilEDR enables attackers to run scripts or binaries either manually or automatically from files uploaded to the target system or stored in the EDR server's library, typically via the live response console or automated response actions. It often supports PowerShell – native to Windows OS – or other scripting languages when the necessary interpreters are available. Attackers can configure EvilEDR to allow unsigned script execution.

We tested various custom and off-the-shelf malware to assess if EvilEDR's activities could be detected or blocked by enterprise EPP/EDR. When executing known-bad tools and techniques, this was usually the case. For instance, we attempted to perform DLL injection using `mavinject.exe` [85], but Windows Defender EPP blocked the operation, and Enterprise EDR generated an alert. We also tested EvilEDR's ability to run a custom ransomware simulation PowerShell script that encrypts a specified folder and its contents. Although the script was saved in the EvilEDR server's library, the EvilEDR agent copied it to its restricted upload folder before execution, which could potentially expose the script to detection by enterprise EPP or EDR, especially if it matches known malicious patterns. However, *our test of running a custom script was successful and remained undetected.* In our tests, MDE and Elastic supported running scripts or executables from uploaded files or the EDR server's library. Sophos lacked built-in file upload and custom script capabilities for response actions, while Trend Micro supported custom scripts but not file uploads.

Persistence. EDR agents are inherently persistent, designed

[6] https://ludus.cloud/
[7] https://attack.mitre.org/techniques/T1190/

Table 1: Summary of EvilEDR's functionalities facilitating key MITRE ATT&CK techniques and whether they are supported by the tested EDR solutions. *Legend:* ✓/✗ indicates whether the functionality is supported/not supported by the corresponding EDR (M - MDE, E - Elastic, S - Sophos, T - Trend Micro)

Tactic	**(Sub)-technique**	**EvilEDR Functionality**	**Result**			
			M	***E***	***S***	***T***
Execution	Command and Scripting Interpreter	Executes scripts and binaries via live response or automated actions	✓	✓	✗	✓
Persistence	Boot or Logon Autostart Execution	Provides inherent persistence; tamper protection inhibits removal	✓	✓	✓	✓
Privilege Escalation	Local Accounts	Allows SYSTEM-level permissions via live response	✓	✓	✓	✓
Defense Evasion	Disable or Modify Tools	Impairs defenses; Replaces EPP by registering its own as the default	✗	✓	✓	✓
	Indicator Blocking	Impairs defenses; Blocks telemetry and response via host isolation	✓	✓	✓	✓
Credential Access	OS Credential Dumping	Extracts SAM and SYSTEM hives via live response	✓	✗	✗	✗
Discovery	System Discovery	Passively collects telemetry on processes, users, and network	✓	✓	✓	✓
	Account Discovery	Collects local and domain accounts through telemetry	✓	✓	✓	✓
Lateral Movement	Lateral Tool Transfer	Uploads tools/files via live response, bypassing MotW	✓	✓	✗	✗
Collection	Data from Local System	Retrieves files and system info via live response or investigation packages	✓	✓	✗	✗
	Automated Collection	Automates data collection via scheduled tasks, event triggers, or APIs	✓	✓	✓	✓
Command & Control	Encrypted Channel	Maintains encrypted C2 channel via live response	✓	✓	✓	✓
Exfiltration	Exfiltration Over C2 Channel	Exfiltrates files via the built-in *get file* command	✓	✓	✗	✓
Impact	Inhibit System Recovery	Prevents recovery because of tamper protection; facilitates system destruction	✓	✓	✓	✓

to remain active on endpoints through reboots, shutdowns, and updates. Additionally, their built-in self-protection mechanisms make them difficult to remove. Attackers leveraging EvilEDR take advantage of this persistence to maintain control over the target system and, potentially, hinder system recovery (as we cover in Section 4.9). It is important to note that EDR solutions are not designed for stealthiness; they are visible in system processes. However, their low performance impact often allows them to go unnoticed unless specifically investigated. In our tests, all tested EDR solutions remained persistent and active.

Privilege Escalation. EDR agents operate with SYSTEM-level privileges. Attackers can use the EvilEDR live response console to execute commands with these privileges without relying on tools like *PsExec* [57], which may trigger alerts. This is particularly useful when the compromised account has limited privileges, such as only allowing software installation. It also provides attackers with the highest level of access for performing critical system-level operations. In our tests, all tested EDR solutions live response consoles run as *NT AUTHORITY\SYSTEM*.

4.2 Defense Evasion

EvilEDR demonstrates significant potential for defense evasion, particularly by impairing defenses. It allows attackers to disable enterprise EPP by registering its own as the default EPP, and block indicators and telemetry to Enterprise EDR through host isolation.

EPP Takeover. EDR vendors provide solutions either as standalone EDR or bundled with EPP capabilities. Attackers using EvilEDR can exploit this setup by overwriting or replacing an existing EPP solution on the system. This can be achieved by manually installing the EPP service or using the EvilEDR server to register its own EPP as the system's default EPP (e.g., Elastic[8], Sophos[9], Trend Micro[10]). When the agent registers as the default EPP, it can disable or replace the existing EPP's functionality. This operation often bypasses detection since registration is a legitimate part of EDR functionality. Once registered, attackers can disable protection features or switch to audit mode, effectively neutralizing the EPP. However, this type of registration is restricted to software that meets specific integration requirements within the Windows Security Center API [56]. During our tests, EvilEDR successfully registered itself as the default EPP in most cases, taking over Windows Defender EPP protections without triggering detection or alerts. No user-facing notifications were generated, and enterprise security teams would likely remain unaware unless they actively monitor the registration entry. However, in cases where a third-party EPP was already installed, EvilEDR caused a dual-registration state, resulting in two active EPPs on the system. This means the Enterprise EPP will still be active, increasing the risk of detection.

Block Telemetry. Host isolation is a key EDR feature that limits a compromised host's communication exclusively to the EDR server. Attackers exploit this to impair defenses by isolating the host, ensuring it communicates only with the EvilEDR server. This prevents the Enterprise EDR agent from sending logs or performing remote response actions.

[8] https://www.elastic.co/guide/en/security/current/configure-endpoint-integration-policy.html

[9] https://support.sophos.com/support/s/article/KBA-000002156

[10] https://docs.trendmicro.com/en-us/documentation/article/trend-vision-one-security-solutions-auto-uninstall

In our tests, *isolating the hosts using EvilEDR caused the Enterprise EDR server to show the host as offline, with no further logs received*. This action was not flagged as malicious and could easily be mistaken for a host simply being offline in enterprise environments. It should be noted that a user of the system will most likely notice the disruption of network traffic. Therefore, an attacker should properly time their use of this feature. By leveraging the system information gathered, as discussed in Section 4.4, the attacker can better understand the usage patterns of the target machine. In our tests, all tested EDR solutions offered a host isolation feature.

4.3 Credentials Access

OS credential dumping is one of the credentials access techniques used to extract passwords and account details from compromised systems. We tested EvilEDR's ability to extract the Security Account Manager (SAM) and SYSTEM registry hive files, which contain critical local and domain account information. In control tests without EvilEDR, OS protections blocked copy operations of these hive files, even with SYSTEM privileges via PsExec [57]. However, *EvilEDR successfully extracted these files through its live response console, and the extraction process went undetected by Enterprise EDR*. Notably, among the tested EDRs, MDE was the only solution capable of bypassing OS restrictions to retrieve these files.

4.4 Discovery

EvilEDR's built-in telemetry and continuous monitoring allow attackers to gather critical information about compromised systems and their environments. This is particularly valuable for both passive and active *system discovery* and *account discovery* without triggering detections. In our tests, all tested EDR solutions collected sufficient telemetry data to provide attackers with valuable insights into systems and accounts. However, the granularity and depth of the telemetry varied across EDR products.

System Discovery. Attackers can use EvilEDR's monitoring capabilities to gain deep visibility into compromised systems [74]. This includes tracking user behavior, system processes, and network traffic without triggering security alerts. By passively collecting this data, EvilEDR helps attackers maintain control, identify opportunities for lateral movement, and blend in with normal system behavior, making detection harder as they expand their foothold.

Account Discovery. Attackers often use account discovery to enumerate local and domain accounts on a compromised machine, typically as a step toward lateral movement. While tools like SharpHound [11] are commonly used for active domain enumeration, our tests indicate that deploying these tools directly via file upload and the response console is likely to trigger detection. In contrast, *EvilEDR leverages its built-in telemetry to passively gather account information without raising alerts*, providing attackers with up-to-date data on both local and domain account usage on the system, all without triggering alerts. It should be noted that the passive enumeration of accounts, while stealthier, will lack the depth and comprehensiveness that tools such as SharpHound offer.

4.5 Lateral Movement

During active attack campaigns, attackers often upload additional tools, ranging from reconnaissance utilities to exploits. These tools can be transferred between compromised devices (lateral tool transfer) or used to spread malicious documents with embedded macros. *EvilEDR allows the upload of arbitrary files*. Notably, files uploaded this way bypass the Mark of the Web (MotW) [91], which flags Internet-downloaded files for extra scrutiny and blocks macros from running in applications like Microsoft Excel and Word. However, if these files are known malicious artifacts, they may still trigger detection by enterprise EPP/EDR when written to the disk or executed on the system. During testing, we used the EvilEDR live response console to upload malicious documents, and Enterprise EDR did not flag or alert on these uploads. In our tests, MDE and Elastic provided an upload feature, whereas Sophos and Trend Micro lacked this functionality.

4.6 Collection

Extending from discovery, EvilEDR enables attackers to actively collect *data from the local system* either manually or through *automated collection*. This functionality is available in all tested EDRs, including MDE[11], Elastic[12], Sophos[13], and Trend Micro[14].

Data from Local System. EvilEDR facilitates the collection of system data through its live response console, enabling direct access to files, processes, and system information. Additionally, it supports built-in investigation package collection, allowing attackers to obtain a comprehensive snapshot of the target system.

Automated Collection. EvilEDR supports automated data collection mechanisms, allowing attackers to schedule periodic data collection, trigger collection in response to specific events, or use APIs. This capability enables attackers to maintain an up-to-date snapshot of target systems, automate collection for large-scale attack campaigns, and coordinate data

[11] https://learn.microsoft.com/en-us/defender-endpoint/respond-machine-alerts

[12] https://www.elastic.co/guide/en/security/current/get-file-api.html

[13] https://docs.sophos.com/central/customer/help/en-us/ManageYourProducts/ThreatAnalysisCenter/LiveDiscover/index.html

[14] https://docs.trendmicro.com/en-us/documentation/article/trend-vision-one-create-ir-collection-playbooks

collection with other malicious operations to achieve their objectives.

4.7 Command and Control

EvilEDR enables C2 through its remote command execution feature in the live response console, enabling attackers to maintain control and persistent access to compromised systems. This way, attackers can remotely execute commands, exfiltrate files, upload and run files, and perform system actions with the same privileges as the EDR agent, often running as SYSTEM, granting extensive control without relying on additional exploitation tools. In our tests, these actions were successfully executed without triggering alerts from Enterprise EDR. All tested EDR solutions had a live response console capable of executing arbitrary commands. Furthermore, EvilEDR live response traffic is encrypted, giving attackers an *encrypted channel* that complicates inspection. However, it is important to note that EvilEDR's C2 stealthiness depends on its hosting configuration. When EvilEDR uses a cloud-hosted option, the EDR agent's traffic is routed to legitimate vendor domains and IPs, significantly reducing the likelihood of detection. For example, MDE agents communicate with trusted Microsoft URLs and services [54], making their traffic appear legitimate and unlikely to alert enterprise security. On the other hand, in self-hosted setups, the agent communicates with an arbitrary domain or IP owned by the attacker, which is more likely to raise suspicions or trigger detection. This applies not only to EvilEDR C2 but also to its general agent communication.

4.8 Exfiltration

Data exfiltration is a primary objective for many threat actors, whether for espionage, ransomware, or expanding their foothold in a network by obtaining sensitive files like private keys[15]. *EvilEDR's file download functionality via its live response console allows attackers to locate and exfiltrate valuable files*. When extraction is initiated, the built-in *get file* command encrypts the files on the host with a password and provides them as downloads via the attacker's browser, effectively operating as *exfiltration over a C2 channel*. During testing, we successfully exfiltrated documents, critical system files, and browser cookies without detection by Enterprise EDR. In our tests, all tested EDR solutions, except Sophos, provided built-in functionality to retrieve files.

4.9 Impact

EvilEDR enables attackers to manipulate, interrupt, or destroy target systems and data. By leveraging remote script execution, persistence, and live response capabilities with elevated permissions, it provides attackers with extensive control to execute attacks. This applies across all EDR solutions we tested. Among its many attack potentials, one technique stands out: *inhibit system recovery*.

Inhibit System Recovery. EvilEDR can exploit EDR's tamper protection to prevent unauthorized removal. Even when EvilEDR is detected, enterprise security cannot uninstall it without first disabling tamper protection. This makes system recovery extremely complex, often leaving OS clean installation or safe boot (which is not always practical) as the sole options. Unlike multi-step approaches attackers might use to achieve similar goals, EvilEDR inherently provides this capability, reducing the risk of detection by Enterprise EDR during execution. In addition to EvilEDR's live response console assisting in data manipulation and encryption (e.g., ransomware), EvilEDR's SYSTEM-level privileges enable attackers to overwrite critical system data and files, rendering them irrecoverable even with forensic tools. For instance, attackers can erase critical system logs or overwrite registry hives to disrupt forensic investigations and recovery efforts.

5 Evaluation

In this section, we evaluate EvilEDR's feasibility and effectiveness in a real-world use case. Feasibility is assessed based on whether EvilEDR can be directly leveraged during a specific attack stage to achieve the intended objective or indirectly assist in a sequence of actions. Effectiveness is assessed by determining whether these actions are prevented or detected by Enterprise EDR. To perform this evaluation, we compare EvilEDR to the actions of a real-world use case. It is important to note that our focus on effectiveness is specifically on the Enterprise EDR's ability to prevent or detect (e.g., trigger alerts) malicious actions, not merely its ability to collect telemetry on those actions. While Enterprise EDRs typically provide comprehensive visibility and telemetry for activities performed on the target system, this does not guarantee that the EDR system will alert those activities as potential threats.

For the assessment, the MITRE Enginuity ATT&CK Evaluations[16] provides an ideal platform. This resource evaluates different security solutions against known adversaries and shares the results with the community. We have chosen the Turla (2023) [60] case that assesses the ability of 29 EDR solutions to detect the tactics, techniques, and procedures (TTPs) used by the Turla APT in one of their campaigns. This evaluation is particularly relevant to our research as it focuses on EDR solutions, including the four EDR solutions covered in this study (MDE [34], Elastic [26], Sophos [81], and Trend Micro [87]), which demonstrated high detection coverage scores during the evaluation. The evaluation consists of two closely-linked scenarios: *Carbon* and *Snake*. The *Carbon* scenario focuses on establishing a foothold in the target network

[15] https://attack.mitre.org/techniques/T1041/

[16] https://attackevals.mitre-engenuity.org/

through spearphishing, with the ultimate goal of creating a watering hole on a web server within this network. The *Snake* scenario builds on Carbon by using the watering hole to target a high-value victim. Once the victim is compromised, Turla moves through their network to an Exchange server, where sensitive information is exfiltrated. Together, these scenarios create a comprehensive evaluation case study.

To evaluate EvilEDR utility, we follow the steps used in Turla's evaluation [60] for each scenario. For clarity and simplicity, we consolidate some steps. In the MITRE evaluation, the evaluator executes specific attack techniques assessing whether the EDR detects them. We use these results as a *baseline*. Then, we employ EvilEDR to execute similar techniques and verify whether they are detected by Enterprise EDR. For a meaningful comparison, we use the same testbed described in Section 3 (resembling a secure enterprise environment) with both enterprise EPP and EDR in block mode. To ensure fairness, we conducted 12 tests for the four EDR solutions covered in this study (MDE, Elastic, Sophos, and Trend Micro), testing each as both Enterprise EDR and EvilEDR. Table 2 reports the results of this comparison. Note that the MITRE evaluation was conducted in 2023. Since then, the EDR vendors may have improved their products regarding coverage and detection techniques, which may convert some crosses into ticks in the corresponding baseline (B) columns.

5.1 Carbon

This scenario follows Turla's multi-step approach to create a watering hole [40] for persistence on a victim's network and to enable further compromise [60].

Initial Compromise & Establish Initial Access. In the baseline scenario, initial access is established through a spearphishing email, tricking the user into downloading and executing a fake software installer that delivers the EPIC backdoor[17]. All four Enterprise EDRs detected this activity as phishing, a known initial compromise technique.

In contrast, EvilEDR indirectly facilitates initial access. Unlike the baseline, the download and installation of all EvilEDR agents remained undetected by Enterprise EDRs, due to its trusted status and signed binary. This demonstrates that EvilEDR is a feasible and effective alternative for establishing initial access. However, it is important to note that the user must have local admin rights to install EvilEDR, as assumed in Section 3.3.

Discovery and Privilege Escalation. Once initial access is achieved, the compromised workstation is enumerated to gather information about the host, including registry details, network configurations, and local and domain accounts (as the workstation is Active Directory (AD) joined). In the baseline scenario, a weak registry permission discovered in a VPN service was exploited to achieve privilege escalation, resulting in local SYSTEM privileges. All four Enterprise EDRs detected these activities as discovery and privilege escalation techniques.

In contrast, EvilEDR directly enables passive and active discovery through its built-in telemetry and live response console. All tested EvilEDRs could passively enumerate the workstation and obtain similar information (including domain accounts) without detection. Additionally, EvilEDR operates on the workstation with SYSTEM privileges and allows attackers to leverage these privileges through its live response console. Unlike the baseline scenario, EvilEDR inherently provides these privileges without exploiting vulnerabilities. All tested EvilEDRs provide a live response console and were not detected by Enterprise EDRs during regular operation, demonstrating EvilEDR as an effective alternative for discovery and privilege escalation. However, it is crucial to note that active enumeration using known malicious commands through the response console still resulted in detection by Enterprise EDR.

Persistence. In the baseline scenario, a second-stage malware, *CARBON-DLL*, is installed to achieve a persistent foothold on the first workstation to facilitate C2 operations and lateral movement in the network. This activity was detected by all four Enterprise EDRs as common malware techniques were used in both the installation and execution of the malicious implant. In contrast, the EvilEDR agent is designed to maintain persistence through system changes such as reboots and updates, activating immediately upon installation. This ensures that EvilEDR starts early during the system's boot process. Although Enterprise EDRs recorded telemetry of EvilEDR's persistence mechanisms, such as the creation of new services, they did not flag these activities as malicious. This demonstrates that EvilEDR is a feasible and highly effective option for maintaining persistence.

Lateral Movement to Domain Controller and Second Workstation. After establishing persistence, the baseline scenario proceeds with password spraying to compromise a domain administrator account. This allows the installation of the *CARBON-DLL* implant on the Domain Controller (DC). From the DC, a second workstation is identified, and credentials are extracted using Mimikatz [63]. These credentials are then used to move laterally to the second workstation. All these activities were detected by Enterprise EDRs as lateral movement techniques. In contrast, EvilEDR can achieve lateral movement through multiple approaches. First, all EvilEDRs can indirectly assist in this process through their live response console by executing custom commands and scripts that replicate the baseline flow. However, this activity is detected by all Enterprise EDRs. Secondly, MDE as EvilEDR can bypass OS protections and export SYSTEM, SAM, and SECURITY registry hives. This provides access to NTLM hashes from local and domain accounts. This activity was not detected by any Enterprise EDR. This demonstrates that EvilEDR can

[17] https://attack.mitre.org/software/S0091/

Table 2: The results of the detection by Enterprise EDRs of the attack techniques executed using default Turla's TTPs (B - Baseline) and using EvilEDR (M - MDE, E - Elastic, S - Sophos, T - Trend Micro). *Legend:* ✓ indicates that the corresponding technique was detected by the corresponding Enterprise EDR (Ent. EDR)

Attack Technique	**Ent. EDR: M**				**Ent. EDR: E**				**Ent. EDR: S**				**Ent. EDR: T**			
	B	**EvilEDR**			**B**	**EvilEDR**			**B**	**EvilEDR**			**B**	**EvilEDR**		
		E	**S**	**T**		**M**	**S**	**T**		**M**	**E**	**T**		**M**	**E**	**S**
Carbon																
Initial Compromise & Establish Initial Access	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Discovery and Privilege Escalation	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Persistence	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Lateral Movement to DC and Second Workstation	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓
Credential Access and Lateral Movement to Web Server	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Installation of Watering Hole	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Snake																
Initial Compromise & Rootkit Installation	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗
Workstation Discovery	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Lateral Movement to Exchange Server	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓
Discovery and Data Exfiltration	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✗	✗	✓	✗	✗	✓

ensure the success of this step, but only if MDE is used.

Credential Access on Admin Host & Lateral Movement to Web Server. Once access to the second workstation is gained, the baseline scenario uses a custom keylogger to harvest credentials. These credentials are then used to move laterally to the Linux Apache web server. All Enterprise EDRs except Elastic detected the use of the keylogger. In contrast, EvilEDR can indirectly facilitate the creation of a custom keylogger script or uploading to the target system using its live response console. However, this results in the same detection pattern, as the modus operandi of uploading and installing a keylogger through EvilEDR is similar to the baseline. This makes EvilEDR a feasible but ineffective alternative for this step.

Installation of Watering Hole. As the final step in the Carbon scenario, the baseline scenario establishes a watering hole via Javascript after gaining access to the web server. Similar to the keylogger, the installation of the watering hole is highly circumstantial and depends on specific adversarial tradecraft. All Enterprise EDRs except Elastic detected the installation of the watering hole. In contrast, using MDE or Elastic's built-in file upload functionality as EvilEDR, the necessary files to install the watering hole can be uploaded to the web server. However, this results in the same detection rate as the baseline. Notably, Sophos and Trend Micro cannot upload custom files directly. Nonetheless, files can still be uploaded through other means, such as hosting them on an attacker-controlled storage and triggering the installation via the live response console. This makes EvilEDR a feasible but ineffective alternative for installing the watering hole.

5.2 Snake

This scenario builds on the Carbon scenario by targeting high-value entities through typo-squatting a website [60]. Once a foothold in the network is established, lateral movement to an Exchange server is carried out, where emails are exfiltrated.

Initial Compromise & Rootkit Installation. Initial compromise to a new target is obtained through the watering hole set up in the Carbon scenario. In the baseline scenario, the new target is enticed to download and run an Adobe Flash installer bundled with the *EPIC* malware. After the initial compromise, the *SNAKE* rootkit is installed to achieve SYSTEM privileges. All Enterprise EDRs detected this phase of the attack. In contrast, assuming local administrative privileges are already present upon initial compromise, EvilEDR indirectly achieves the initial compromise by being packaged with other software or by enticing the user to install an EDR agent directly. As with the Carbon scenario, Enterprise EDRs did not detect the installation of EvilEDR. Since EvilEDR inherently operates with SYSTEM privileges, it eliminates the need for a rootkit to achieve such privileges, demonstrating that EvilEDR is a feasible and effective alternative for establishing initial access in the Snake scenario.

Workstation Discovery. In the baseline scenario, discovery on the target workstation revealed a file server and the file server admin user. The scenario progresses by impersonating a low-privileged user to enumerate mapped drives, which reveals that the user's home directory is mapped to the file server. All Enterprise EDRs detected the enumeration behavior. In contrast, EvilEDR, as with Carbon, leverages passive built-in telemetry to provide some information for enumeration. However, active discovery using the live response console is required in all EvilEDRs to discover the file system structure and network locations. This activity was successfully detected by all Enterprise EDRs. This demonstrates that EvilEDR is a feasible alternative for workstation discovery. However, since all necessary telemetry is not passively available for this specific step, it is ultimately ineffective.

Lateral Movement to Exchange Server. This stage of the Snake scenario involves several steps of lateral movement. The baseline scenario begins with lateral movement from the initial target workstation to a newly discovered file server by running PsExec to execute another copy of the *SNAKE* rootkit. From there, a pass-the-hash attack is executed after dumping credentials on the file server using *Mimikatz*. This provides access to an admin workstation. This access ultimately enables lateral movement to an Exchange server. In the baseline scenario, all Enterprise EDRs successfully detected these lateral movement techniques. In contrast, all EvilEDRs can indirectly support these steps through telemetry collection and command execution. Additionally, lateral movement steps can be substituted with the installation of EvilEDR on the target machine. However, with the exception of MDE, all EvilEDRs require external tools, such as Mimikatz, to obtain the necessary credentials. These actions were detected similarly to the baseline scenario. This demonstrates that EvilEDR is a feasible option for executing these steps but, with the exception of MDE, an ineffective alternative for carrying out these specific lateral movement operations.

Discovery and Data Exfiltration. In the baseline scenario, a C2 channel is established using the LightNeuron implant. LightNeuron utilizes email as a C2 mechanism by attaching JPG images with embedded encrypted commands via steganography. Eventually, it automatically collects all emails containing specific recipients into a log file, then exfiltrated over the existing C2 channel. All Enterprise EDR vendors successfully detected at least one part of the exfiltration stage. In contrast, EvilEDR solutions like MDE, Elastic, and Trend Micro can exfiltrate email traffic using their built-in *get-file* features. Additionally, these agents can be configured with custom response actions to automate the exfiltration process. Notably, none of Enterprise EDRs detected these activities, making EvilEDR, except for Sophos, a feasible and effective alternative for exfiltrating email data.

6 Defense

This section presents strategies to protect against EvilEDR at enterprise and vendor levels. It is not an exhaustive guide to stopping this threat. Instead, it aims to equip enterprises with actionable measures to prevent and detect EvilEDR and guide EDR vendors on limiting the misuse of their software.

6.1 Enterprise-Level Defenses

We approach enterprise defense from a multi-layered perspective, combining preventive and detective controls. For EvilEDR, preventive controls aim to block its deployment and execution, while detective controls focus on identifying its presence after deployment. Our guiding principle is that *every EDR is a potential EvilEDR, except those explicitly deployed and controlled by the enterprise*. Enterprises may choose to deploy a single EDR, combine multiple EDR solutions [14], or opt not to deploy any EDR. However, it is essential to note that a machine cannot have multiple instances of the same EDR solution from the same vendor, even of different versions. This effectively narrows detection efforts to unauthorized EDR instances while ensuring clarity for authorized deployments.

6.1.1 Prevention

We focus on post-exploitation controls, assuming attackers have already gained initial access, bypassing early-stage preventive measures. While not exclusively tailored to EvilEDR, these measures effectively hinder its deployment and execution. Because EvilEDR requires elevated privileges to install, enterprises should enforce the *principles of least privileges*, where standard users must not have local admin rights and administrative privileges should be assigned to non-personal accounts managed through Privileged Access Management (PAM) solutions. When properly implemented and monitored, this approach significantly increases the difficulty for attackers, requiring them to exploit privilege escalation vulnerabilities or compromise the PAM system to acquire valid credentials. For scenarios where administrative credentials are compromised, or attackers escalate privileges, enterprises should enforce *strict software installation policies*, for example, using Windows Defender Application Control (WDAC) or Software Restriction Policies (SRPs) Group Policy [50]. This enforces a strict allowlist where only vetted and authorized applications can be installed or executed on endpoints, even if they appear legitimate. Additionally, enterprises should implement *restrictive network policies* which prevent an EvilEDR from communicating with its management server. Enterprises must apply strict outbound controls on critical endpoints, allowing communication only with approved domains and IP addresses. Although these controls are more challenging on user endpoints, network security solutions (e.g., proxies) can still monitor and block suspicious connections indicative of EvilEDR activity (for instance, traffic directed to an external EDR domain the organization does not use).

6.1.2 Detection

In this section, we cover methods for detecting EvilEDR in action. EDR solutions use both drivers and processes to operate. Drivers enable EDR to function at the kernel level and require explicit approval from Microsoft, which involves a strict validation process and registration of drivers [51]. This approval process limits kernel access to a small set of approved vendors and prevents unauthorized EDR solution creation. On the other hand, EDR processes run in user space and serve as the EDR client to communicate with the management server. These processes are typically instantiated from signed binaries, adding further restrictions. Notably, *these drivers and processes are protected by tamper protection and signing mechanisms, ensuring that any modification, renaming, or unauthorized changes are flagged as EDR tampering or evasion attempts*. These constraints facilitate rule-based detection to identify all EDR drivers and processes while excluding those explicitly authorized and deployed by the enterprise. In our detection approach, we include both EDR drivers and processes, as we observed that some open-source EDR solutions often lack kernel drivers and rely solely on signed processes. Table 3 provides an overview of EDR solutions, including their type, drivers, and processes. The list was compiled through experimentation, desk research, and cross-checked using multiple sources [30, 39, 51, 66, 76, 90].

Implementation: System activity often leaves traces in system logs, with the Windows Event Log being a key resource for identifying malicious behavior [19]. Monitoring event IDs, such as those for driver loading and process creation, can indicate signs of EvilEDR. However, the implementation depends on the enterprise's specific context, including available tools such as EDR or Security Information and Event Management (SIEM) solutions and the formats supported by those tools. To ensure our detections' adaptability and broad applicability, we provide detection rules in Sigma[18] format - allowing detections to be shared in a common language and translated into tool-specific rules. In environments without SIEM or EDR solutions, Sigma rules can be converted into scripts or queries compatible with native tools, such as Windows Event Viewer, PowerShell, or other log parsers. Implementing detection within the broader enterprise context is essential to understand which EDR solutions are legitimate and distinguish authorized activities from potential intrusions, reducing false positives and maintaining effective security operations.

EDR Driver Load: We create detection rules based on driver load events, which serve as an early indicator of EvilEDR activity. EDR solutions install their own drivers with unique names and are typically located in the system's driver directory. *Sysmon Event ID 6* logs all driver load events. Using these events, detection rules can identify all EDR drivers while excluding those explicitly authorized and deployed by the enterprise. This approach minimizes false positives by narrowing detections to unauthorized EDR drivers, providing a high-confidence signal for detecting EvilEDR activity. Although we aim to detect EvilEDR during deployment, this rule can also identify previously loaded unauthorized drivers if historical logs are available for retrospective analysis. We provide a general detection rule in Sigma format, along with specific rules for each EDR solution covered in this study, written using MDE's Kusto Query Language (KQL), Elastic's Elastic Query Language (EQL), Sophos's SQL-based search query, and Trend Micro's custom filters [3]. Note that these rules should still be tailored to the specific enterprise setup.

EDR Processes: Similar to detecting EDR driver loads, we detect EDR processes to find solutions that do not utilize drivers (e.g., open-source EDRs). Our approach focuses on EDR process names rather than full paths or hash values. Process names are typically fixed by the vendor, with minimal likelihood of change. Attackers' attempts to modify these names would trigger tamper protection mechanisms and be unable to change process names. While there is a small risk of false positives if another legitimate process shares the same name, this method is more practical and reliable than using full paths or hash values. Full paths can differ depending on system installation paths, making them unreliable for consistent detection. Hash values can change due to vendor updates, requiring the tune of detection rules to remain effective. Therefore, using process names strikes a balance between accuracy and maintainability. During detection rule rollouts, if security teams encounter legitimate processes with the same name as an EDR process, they can adjust the rule by excluding the conflicting name or modifying the detection logic to fit their environment. These adjustments are typically a one-time effort to ensure accurate detection while minimizing false positives. We share the detection rules for unauthorized EDR processes built using Sigma, MDE's KQL, Elastic's EQL, Sophos's SQL-based query, and Trend Micro's custom filters [3]. As noted before, enterprise defenders should tailor these rules to exclude authorized EDR processes.

Manual Investigation: In environments without a central log source or in situations where attackers have cleared logs to hide driver loads, security teams can manually check for the presence of EvilEDR. This can be done using *PowerShell* to detect EDR processes and drivers. This approach allows security teams to use endpoint management tools like Intune to centrally execute scripts on endpoints. Additionally, it enables them to adapt the script's logic as needed, for example, to trigger additional workflows or perform remediation actions. To support this, we include PowerShell scripts as artifacts [3]. These artifacts cover detecting EDR driver files and identifying running EDR processes, and they do not require administrative privileges to run.

6.2 Vendor-Side Hardening

EDR vendors can also adopt measures to limit the misuse of their software. The first approach is particularly effective when an enterprise already has an EDR deployed. EDR vendors can incorporate built-in notifications that alert when another EDR solution installation is detected, automatically blocking it by default. This mechanism ensures that the enterprise security team explicitly approves any attempted deployment of EvilEDR. Enterprise EDR can actively monitor and block conflicting installations, making it an effective defense in environments with an active EDR. Additionally, this approach shifts the responsibility to vendors to maintain an accurate, up-to-date list of EDR solutions. On the other hand,

[18] https://github.com/SigmaHQ/sigma

this approach may require significant architectural changes to EDR software and close collaboration among vendors. Another approach, accommodating enterprises without an EDR, is restricting access to vendors' solutions by implementing stricter processes, such as background checks or manual verifications for access or trial licenses. During our experiments, we observed that some EDR vendors already employ manual verification for trial licenses. However, this method has limitations, as sophisticated threat actors can bypass these checks by impersonating legitimate organizations. Additionally, overly restrictive measures may reduce EDR availability, hindering researchers' ability to test and evaluate EDR solutions. As a last resort, in the absence of an Enterprise EDR, endpoint management tools like Intune could theoretically be used to notify administrators of unapproved EDR installations.

7 Discussion and Limitations

EDR developers' aspiration to make their solutions powerful in combating digital threats also sharpened the other side of the "double-edged sword," allowing EDR functionality to be used in offensive scenarios. To our knowledge, we are the first to present a study on how an EDR system, widely used by enterprises to protect their endpoints, can be repurposed into an offensive tool.

7.1 Discussion

From an adversarial view, repurposing offers benefits in four key areas: ease of deployment and configuration, integration of attack functionalities, operation and persistence, challenges for defenders and vendors.

As we discussed in Section 3, the configuration and deployment of an EvilEDR is relatively simple. EDR vendors often provide detailed documentation and automated deployment scripts, making the process straightforward [31, 52, 82]. Attackers do not need detailed knowledge of the existing EPP/EDR, as EvilEDR can operate alongside them.

EvilEDR integrates multiple attack functionalities typically performed by specialized tools, demonstrating the extent to which an EDR system can enhance an attacker's capabilities. In Section 4, we demonstrate EvilEDR's attack potential, and in Section 5, we showcase its effectiveness through real-world scenarios. To impair defenses, EvilEDR can facilitate an EPP takeover by registering its own EPP, which attackers can deliberately weaken to render it ineffective. Additionally, it allows attackers to leverage its built-in host isolation feature to block Enterprise EDR telemetry and response. For credential dumping, EvilEDR replicates some functionality of Mimikatz [63], extracting SAM and SYSTEM registry files, but with a lower detection risk due to its trusted status. For C2, its remote command execution capabilities parallel those of Cobalt Strike[19], enabling continuous control over compromised systems without relying on external C2 frameworks. EvilEDR's scripting and automation features facilitate lateral movement within the network, similarly to Empire[20], while maintaining the appearance of legitimate processes. Elevated system privileges allow EvilEDR to substitute tools like PsExec [57] for accessing system services[21], ensuring its operations persist even after a system reboot, akin to PlugX[22], but without triggering alerts associated with backdoors.

After installation and initial configuration, EvilEDR can operate unnoticed, which is one of its most dangerous aspects. It also ensures long-term access by resisting standard maintenance actions like system reboots. We observed no significant changes in system resource utilization – memory, CPU, or disk I/O – during or after EvilEDR installation. In our experiments with various free and paid EDR solutions, EvilEDR did not affect user experience or system performance, maintaining a low profile [36]. From a network security monitoring perspective, EDR traffic is typically encrypted using standard protocols like HTTPS, making it difficult for analysts to distinguish between legitimate telemetry data and obfuscated malicious commands. In cloud-hosted EvilEDR setups, traffic generally appears benign, as it communicates with recognized vendor domains. On the other hand, self-hosted EvilEDR setups route traffic to attacker-controlled IP addresses or custom domains, which may prompt further investigation by security analysts due to the unusual or suspicious destination.

Unlike malicious software, which may be blocked due to the presence of their Indicators of Compromise (IOCs) in threat intelligence lists, enterprise defenders cannot rely on IOCs when EvilEDR is identified in an APT attack campaign, as the EvilEDR used in that campaign could be the Enterprise's own EDR solution. Even when enterprise security identifies EvilEDR, its tamper protection often requires a clean OS reinstall or manual removal in safe boot mode, which is not always effective and operationally challenging at scale. Furthermore, EDR vendors cannot easily address the misuse of their tools since repurposing EDR is not classified as a software vulnerability and cannot be detected with predefined rules, particularly in environments where multiple EDR solutions are in use simultaneously [15]. This presents a significant challenge, as the very features that make EDRs effective for defense also make them potent tools in the hands of attackers. We proposed measures in Section 6.2; however, these measures are not foolproof and might require substantial changes to current architectures.

7.2 Limitations

Our study assumes that the attacker has local admin privileges (described in Section 3.3) to install EvilEDR on the target sys-

[19] https://attack.mitre.org/software/S0154/

[20] https://attack.mitre.org/software/S0363/

[21] https://attack.mitre.org/techniques/T1569/002/

[22] https://attack.mitre.org/software/S0013/

tem. Thus, they must first gain initial access and potentially perform a local privilege escalation. This level of access can be obtained through various means that are out of the scope of this study, such as system vulnerability exploitation, access via compromised accounts, or using social engineering techniques. It is also possible to run endpoint security agents with standard user privileges [28]. However, their functionality is limited in this case and typically does not extend to full EDR capabilities, such as code execution.

EvilEDR relies on network communication with its management console, requiring traffic to its designated targets to be permitted. Network access requirements depend on whether EvilEDR is self-hosted [31] or cloud-hosted. Self-hosted setups require access to the attacker's infrastructure, such as external IP addresses and domains, while cloud-hosted setups need access to the EDR vendor's domains and IPs [53]. In highly restrictive environments, where outbound traffic is filtered to authorized domains and IPs, these limitations can prevent EvilEDR installation or operation.

One significant limitation of using EvilEDR as a malicious tool is its lack of flexibility to support various execution techniques. Unlike dedicated C2 frameworks, designed to be highly adaptable and offer multiple ways to perform the same task, EvilEDR, being legitimate software, typically lacks this versatility. For instance, real C2 frameworks often support various communication protocols and C2 channels to evade detection and maintain persistence [71]. In contrast, EvilEDR would be constrained to the methods and protocols inherent to its legitimate functionality, thereby limiting its effectiveness and adaptability in diverse attack scenarios.

EDR systems are not designed for stealthiness or malicious use, making EvilEDR more likely to leave a noticeable footprint than other C2 frameworks. During deployment, EvilEDR may install a trusted certificate in the Certificate Store and modify the registry, system processes, services, and drivers – changes that could be flagged. However, these are often perceived as legitimate and align with standard enterprise software deployment and maintenance practices, particularly since the agent is trusted and signed. EvilEDR can effectively *"hide in plain sight"* due to its inherent role as a defense tool. In environments, which are not heavily restricted and baselined, an EDR process does not stand out to an analyst. Still, the attacker must be aware of any existing EDR solutions, as deploying an EvilEDR from the same vendor, even of a different version, most likely will not work and could trigger detection if perceived as tampering by the existing EDR.

7.3 Future Research

Several areas remain open for future exploration. First, while studies have examined tampering with EDR systems [8,17,29, 35,41,44,46], future research could focus on leveraging these techniques to transform Enterprise EDR into EvilEDR. This involves taking control of an existing Enterprise EDR deployment and redirecting its communication to a malicious server under the attacker's control. Such an approach could shed light on highly sophisticated, large-scale threats. Notably, this was our initial focus. Despite some progress, we could not fully compromise EDRs due to robust integrity controls. However, deploying EvilEDR alongside Enterprise EDR achieved the same objective of leveraging EDR for malicious purposes. During this process, we discovered a vulnerability in one EDR vendor's product[23] that bypasses its tamper protection, rendering it completely ineffective. We were awarded a bounty for this discovery.

Another area of future research is the development of more advanced real-time detection mechanisms. Rather than relying on static rules targeting EDR drivers and processes, ML techniques could be used to recognize subtle behavioral patterns distinguishing EvilEDR from legitimate EDR activity. These approaches could be tested against various EDR tools to assess scalability and generalizability. Finally, there is potential to develop new mitigation techniques tailored explicitly for EvilEDR. This could include designing automated defenses capable of recognizing and responding to EvilEDR in real time or developing honeypot-style EDR solutions that attract and neutralize malicious activity. The detection strategies outlined in this research could also be tested in real-world scenarios, particularly in multi-EDR environments, where overlapping systems may present unique challenges.

8 Conclusion

In this study, we introduced the concept of EvilEDR – repurposing EDR from a defensive to an offensive tool – a simple, effective, and highly applicable technique for attackers. We demonstrated how the strengths of EDR systems can become a liability for defenders when attackers misuse these systems. We provided a playbook of attack techniques where EvilEDR can be particularly effective, supported by evaluations in real-world scenarios. Once deployed, EvilEDR allows attackers to use built-in capabilities to compromise defenses (e.g., EPP takeover and blocking telemetry via host isolation), gain deep visibility into the target environment, maintain command and control, and exfiltrate data. Given EDR's inherent power, EvilEDR represents a significant threat and poses a serious challenge to enterprise defenders. However, as proposed in this paper, even basic rule-based security measures and EDR hardening can significantly reduce the risk of EvilEDR deployment or operation within enterprise environments. Through this study, we aim to highlight this unexplored attack vector and foster collaboration between enterprises and security vendors to develop more resilient defenses. By addressing the potential misuse of EDR systems, we seek to ensure that these tools are used for their intended purpose: strengthening security rather than undermining it.

[23] Not disclosed as it remains unresolved at the time of writing.

Ethical Considerations

We conducted this research with a commitment to ethical standards. During the work on this project, we discovered a vulnerability that was reported to the respective vendor, adhering to responsible disclosure practices. On January 9, 2025, we received an acknowledgment from the vendor, along with a bounty award. However, since the issue remains unresolved, we have opted not to elaborate on the details. We are committed to ensuring that the vendor has adequate time to address the vulnerability before we provide further details on the discovered issue.

Regarding EDR repurposing, it is important to clarify that this is not a vulnerability in the traditional sense; rather, it is a weakness in the way EDR systems can be used to perform malicious activities. The systems function as designed, but their capabilities can be misused in specific contexts. This highlights a security concern, especially for organizations that use multiple EDR solutions simultaneously. Raising awareness of this potential misuse is crucial, and we believe it is in the public interest to bring attention to this issue. Following the reviewers' recommendations, on December 21, 2024, we reported our findings to the vendors of the EDR systems analyzed in this study. We included a preliminary version of our manuscript, offered to share our expertise, and invited them to provide their input on the matter. At the time of the preparation of the camera-ready version of this paper (January 27, 2025), we have received responses from two vendors acknowledging the issue. One vendor has assessed the issue as low severity, indicating that it does not require immediate servicing. The second vendor is currently investigating various mitigation strategies. Since the responsible disclosure period has not yet concluded, we have withheld the details about the vendors and our interactions with them. Adhering to high ethical standards, we have also placed our work under embargo, giving the vendors sufficient time to investigate the issue and implement appropriate mitigations.

We are sure that all enterprises should be aware about this weakness and take countermeasures to mitigate the risk of its execution, using, for instance, the defense mechanisms proposed in this work. Moreover, to our opinion, the attention of the security community to this issue might drive the OS and EDR vendors to collaborate and add functionality mitigating this risk.

Open Science Policy Compliance

We publicly release all the artifacts developed in this study, so that the scientific community can validate our findings, while organizations can use them to detect the misuse of EDR solutions [3]. As we have discussed earlier, we have made a responsible disclosure of the found vulnerability in one of the EDRs, and shared our findings with the vendors of the EDR systems analyzed within this study.

References

[1] Janus Agcaoili and Earle Earnshaw. Legitimate tools weaponized for ransomware in 2021. https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/locked-loaded-and-in-the-wrong-hands-legitimate-tools-weaponized-for-ransomware-in-2021, April 2021.

[2] AhnLab. Defense evasion techniques detected by AhnLab EDR. https://asec.ahnlab.com/en/63145/, March 2024.

[3] Kotaiba Alachkar, Dirk Gaastra, Eduardo Barbaro, Michel van Eeten, and Yury Zhauniarovich. Artifacts for the paper "EvilEDR: Repurposing EDR as an offensive tool". https://doi.org/10.5281/zenodo.14732733, January 2025.

[4] Riccardo Ancarani. Attacking an EDR: Part 2. https://riccardoancarani.github.io/2023-09-14-attacking-an-edr-part-2/, September 2023.

[5] David Balaban. The role of Endpoint Detection and Response in today's enterprise security. https://www.forbes.com/sites/davidbalaban/2021/12/17/the-role-of-endpoint-detection-and-response-in-todays-enterprise-security/, April 2022.

[6] Guru Baran. Hackers use number of legitimate tools in ransomware attacks. https://gbhackers.com/legitimate-tools-ransomware/, March 2024.

[7] Frederick Barr-Smith, Xabier Ugarte-Pedrero, Mariano Graziano, Riccardo Spolaor, and Ivan Martinovic. Survivalism: Systematic analysis of Windows malware Living-Off-The-Land. *IEEE Symposium on Security and Privacy*, pages 1557–1574, 2021.

[8] Abdul Basit Ajmal, Shawal Khan, and Farhana Jabeen. Defeating modern day anti-viruses for defense evaluation. In *International Conference on Frontiers of Information Technology*, pages 255–260, 2022.

[9] Jonathan Beierle and Logan Goins. Weaponizing WDAC: Killing the dreams of EDR. https://beierle.win/2024-12-20-Weaponizing-WDAC-Killing-the-Dreams-of-EDR/, December 2024.

[10] Ben Bernstein. Remote control: Detecting RMM software and other remote admin tools. https://redcanary.com/blog/threat-detection/rmm-software/, April 2024.

[11] BloodHoundAD. SharpHound. https://github.com/BloodHoundAD/SharpHound, July 2024.

[12] Tiberiu Boros, Andrei Cotaie, Antrei Stan, Kumar Vikramjeet, Vivek Malik, and Joseph Davidson. Machine learning and feature engineering for detecting Living off the Land attacks. In *International Conference on Internet of Things, Big Data and Security*, 2022.

[13] Broadcom. Protection highlight: EDR vs defense evasion. https://www.broadcom.com/support/security-center/protection-bulletin/protection-highlight-edr-vs-defense-evasion, April 2024.

[14] Dan Brown. Effective detection and response. https://www.danbrown.co/effective-detection-and-response/, 2024.

[15] Dan Brown. One EDR vs. multiple EDR: Effective detection and response. https://www.danbrown.co/effective-detection-and-response/, June 2024.

[16] Symantec by Broadcom. Inviting users to download and install the Symantec Agent. https://techdocs.broadcom.com/us/en/symantec-security-software/endpoint-security-and-management/endpoint-security/sescloud/Installing-the-Symantec-Agent-and-enrolling-devices/inviting-users-to-download-and-install-the-symante-v130897009-d4155e5436.html, December 2024.

[17] Efstratios Chatzoglou, Georgios Karopoulos, Georgios Kambourakis, and Zisis Tsiatsikas. Bypassing antivirus detection: old-school malware, new tricks. https://arxiv.org/abs/2305.04149, 2023.

[18] B J Chinmaya, Sujay Arun Kudtarkar, and Mohana. Targeted ransomware attacks and detection to strengthen cybersecurity strategies. In *International Conference on Automation, Computing and Renewable Systems*, pages 1039–1044, 2023.

[19] Cyber5W. Windows event logs analysis. https://blog.cyber5w.com/eventlog-analysis, June 2024.

[20] Cybereason. The timeline to consolidation of Endpoint Protection Platforms (EPP) and Endpoint Detection and Response (EDR). https://www.cybereason.com/blog/the-timeline-to-consolidation-of-endpoint-protection-platforms-epp-and-endpoint-detection-and-response-edr, 2024.

[21] Cybersecurity and Infrastructure Security Agency (CISA). Guide to securing remote access software. https://www.cisa.gov/sites/default/files/2023-06/Guide%20to%20Securing%20Remote%20Access%20Software_clean%20Final_508c.pdf, June 2023.

[22] Cybersecurity and Infrastructure Security Agency (CISA). Protecting against malicious use of remote monitoring and management software. https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-025a, January 2023.

[23] Cymulate. Blindside: A new technique for EDR evasion with hardware breakpoints. https://cymulate.com/blog/blindside-a-new-technique-for-edr-evasion-with-hardware-breakpoints, March 2024.

[24] Cymulate. EDR techniques: How attackers bypass detection. https://cymulate.com/blog/edr-techniques/, February 2024.

[25] Redops Daniel Feichter. A story about tampering EDRs. https://redops.at/en/blog/a-story-about-tampering-edrs, March 2024.

[26] Tamarian Del Conte, James Spiteri, and Paul Ewing. Elastic Security stops memory and kernel attacks in round 5 of MITRE Engenuity Evaluations. https://www.elastic.co/blog/elastic-security-stops-memory-kernel-attacks-round-5-mitre-engenuity-evaluations, 2023.

[27] Kuiye Ding, Shuhui Zhang, Feifei Yu, and Guangqi Liu. LOLWTC: A Deep Learning approach for detecting Living Off the Land attacks. In *International Conference on Cloud Computing and Intelligent Systems*, pages 176–181, 2023.

[28] Elastic. Run Elastic Agent without administrative privileges. https://www.elastic.co/guide/en/fleet/current/elastic-agent-unprivileged.html, 2024.

[29] Osama Ellahi, Munam Ali Shah, and Muhammad Usman Rana. The ingenuity of malware substitution: Bypassing next-generation antivirus. In *International Conference on Automation and Computing*, pages 1–5, 2021.

[30] Microsoft Security Experts. A BlackByte ransomware intrusion case study. https://techcommunity.microsoft.com/blog/microsoftsecurityexperts/a-blackbyte-ransomware-intrusion-case-study/3841810, 2024.

[31] NetByteSec Fareed. Detection engineering part 1: Setting up Elastic, Kibana and Fleet server for SIEM and EDR. https://notes.netbytesec.com/2023/06/install-elastic-kibana.html, June 2023.

[32] Nolan Foster. Best EDR solutions: Top 10 solutions to consider in 2024. https://www.acecloudhosting.com/blog/best-edr-solutions/, March 2023.

[33] Brett Gallant. The evolution of cyber attacks: A decade of change in the US and Canada. https://www.linkedin.com/pulse/evolution-cyber-attacks-decade-change-us-canada-brett-gallant-4acfe/, August 2024.

[34] Tanmay Ganacharya. Microsoft 365 Defender demonstrates 100 percent protection coverage in the 2023 MITRE Engenuity ATT&CK® Evaluations: Enterprise. https://www.microsoft.com/en-us/security/blog/2023/09/20/microsoft-365-defender-demonstrates-100-percent-protection-coverage-in-the-2023-mitre-engenuity-attck-evaluations-enterprise/, September 2023.

[35] Ziya Genç, Gabriele Lenzini, and Daniele Sgandurra. A game of "cut and mouse": bypassing antivirus by simulating user inputs. In *Annual Computer Security Applications Conference*, pages 456–465, 12 2019.

[36] Dave Gruber. The need for speed: Second generation EDR. https://www.fortinet.com/resources/esg-white-paper, May 2020.

[37] Michael Haag. Windows remote access software hunt. https://research.splunk.com/endpoint/8bd22c9f-05a2-4db1-b131-29271f28cb0a/, August 2024.

[38] Matt Hand. How attackers evade your EDR/XDR system — and what you can do about it. https://www.csoonline.com/article/3476179/how-your-xdr-is-evaded.html, July 2024.

[39] HarleyQu1nn. AggressorScripts: EDR detection script. https://raw.githubusercontent.com/harleyQu1nn/AggressorScripts/master/EDR.cna, 2023.

[40] Ruben Jami. Watering hole attack: How it works and how to prevent it. https://cymulate.com/blog/watering-hole-attack-dont-drink-water/, August 2024.

[41] Helvio Carvalho Junior. HookChain: A new perspective for bypassing EDR solutions. https://arxiv.org/abs/2404.16856, 2024.

[42] Kaialogen. EDR evasion techniques. https://kaialogen.com/posts/EDR_Evasion_Techniques, September 2023.

[43] Matt Kapko. SMBs hit by rise in legitimate tool-based attacks. https://www.cybersecuritydive.com/news/smbs-legitimate-tool-attacks/700410/, November 2023.

[44] George Karantzas and Constantinos Patsakis. An empirical assessment of Endpoint Detection and Response systems against Advanced Persistent Threats attack vectors. *Journal of Cybersecurity and Privacy*, 1(3):387–421, 2021.

[45] Bart Lenaerts-Bergmans. What are Living off the Land (LOTL) attacks? https://www.crowdstrike.com/cybersecurity-101/living-off-the-land-attacks-lotl/, February 2023.

[46] Trevor M. Lewis and Bhaskar P. Rimal. Effects of removing user-land hooks in endpoint protection during attack experiments. *IEEE Access*, 12:15820–15844, 2024.

[47] John Leyden. Attackers increasingly using legitimate remote management tools to hack enterprises. https://www.csoonline.com/article/3487743/attackers-increasingly-using-legitimate-remote-management-tools-to-hack-enterprises.html, August 2024.

[48] Etay Maor. Here's how cybercriminals bypass EDR and why security teams need a defense-in-depth approach. https://www.scmagazine.com/perspective/heres-how-cybercriminals-bypass-edr-and-why-security-teams-need-a-defense-in-depth-approach, June 2023.

[49] Greg McDonough. Evolving cyber threats & hacking techniques. https://www.rsaconference.com/library/blog/evolving-cyber-threats-and--hacking-techniques, September 2024.

[50] Microsoft. Software restriction policies. https://learn.microsoft.com/en-us/windows-server/identity/software-restriction-policies/software-restriction-policies, January 2023.

[51] Microsoft. Allocated altitudes. https://learn.microsoft.com/en-us/windows-hardware/drivers/ifs/allocated-altitudes, 2024.

[52] Microsoft. Configure endpoints using scripts in Microsoft Defender for Endpoint. https://learn.microsoft.com/en-us/defender-endpoint/configure-endpoints-script, April 2024.

[53] Microsoft. Configure the environment for Microsoft Defender for Endpoint. https://learn.microsoft.com/en-us/defender-endpoint/configure-environment, June 2024.

[54] Microsoft. Configure your network environment to ensure connectivity with Defender for Endpoint service. https://learn.microsoft.com/en-us/defender-endpoint/configure-environment, October 2024.

[55] Microsoft. Microsoft digital defense report 2024. https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/final/en-us/microsoft-brand/documents/Microsoft%20Digital%20Defense%20Report%202024%20%281%29.pdf, 2024.

[56] Microsoft. Microsoft virus initiative. https://learn.microsoft.com/en-us/defender-xdr/virus-initiative-criteria, May 2024.

[57] Microsoft. PsExec v2.43. https://learn.microsoft.com/en-us/sysinternals/downloads/psexec, March 2024.

[58] Van Mieghem. Blueprint for evading EDR in 2022. https://vanmieghem.io/blueprint-for-evading-edr-in-2022/, April 2022.

[59] Kyle Mistele. A beginner's guide to EDR evasion. https://kylemistele.medium.com/a-beginners-guide-to-edr-evasion-b98cc076eb9a, September 2021.

[60] MITRE ATT&CK. Turla enterprise evaluation 2023. https://attackevals.mitre-engenuity.org/enterprise/turla/, 2023.

[61] MITRE ATT&CK. Valid accounts (Technique T1078). https://attack.mitre.org/techniques/T1078/, March 2023.

[62] MITRE ATT&CK. Phishing (Technique T1566). https://attack.mitre.org/techniques/T1566/, March 2024.

[63] MITRE ATT&CK. S0002: Mimikatz. https://attack.mitre.org/software/S0002/, February 2024.

[64] MITRE Corporation. ATT&CK Matrix for Enterprise. https://attack.mitre.org/, 2024.

[65] Elizabeth Montalbano. Novel EDR-killing 'GhostEngine' malware is built for stealth. https://www.darkreading.com/cyberattacks-data-breaches/novel-edr-killing-ghostengine-malware-stealth, May 2024.

[66] Netero1010. EDRSilencer. https://github.com/netero1010/EDRSilencer, 2024.

[67] Ruolin Ning, Wenjuan Bu, Ju Yang, and Shuang Duan. A survey of detection methods research on Living-Off-The-Land techniques. In *IEEE International Conference on Sensors, Electronics and Computer Engineering*, pages 159–164, 2023.

[68] Brigid O'Gorman. A disturbing trend in ransomware attacks: Legitimate software abuse. https://www.cio.com/article/645393/a-disturbing-trend-in-ransomware-attacks-legitimate-software-abuse.html, July 2023.

[69] Talha Ongun, Jack W. Stokes, Jonathan Bar Or, Ke Tian, Farid Tajaddodianfar, Joshua Neil, Christian Seifert, Alina Oprea, and John C. Platt. Living-Off-The-Land command detection using active learning. In *International Symposium on Research in Attacks, Intrusions and Defenses*, 2021.

[70] OpenText. What is an insider threat? https://www.opentext.com/what-is/insider-threat, 2024.

[71] Jorge Orchilles. C2 Matrix. https://howto.thec2matrix.com/, February 2024.

[72] Pentera. Zero footprint attacks: 3 steps to bypass EDR with reflective loading. https://pentera.io/blog/zero-footprint-attacks-3-steps-to-bypass-edr-with-reflective-loading/, June 2024.

[73] Albert Puah. Dual-intent tools commonly used by hackers and how to defend against them. https://community.ibm.com/community/user/security/blogs/albert-puah/2023/04/05/dual-intent-tools-commonly-used-by-hackers-and-how, April 2023.

[74] Red Canary. Endpoint visibility & EDR: Important assessment criteria. https://redcanary.com/blog/security-operations/how-to-improve-endpoint-visibility-with-edr/, April 2024.

[75] SafeBreach. The dark side of EDR: An offensive tool in the wrong hands. https://www.safebreach.com/blog/dark-side-of-edr-offensive-tool/, April 2024.

[76] Jacob Santos, Cj Arsley Mateo, and Sarah Pearl Camiling. Silent threat: Red Team tool EDRSilencer disrupting endpoint security solutions. https://www.trendmicro.com/en_us/research/24/j/edrsilencer-disrupting-endpoint-security-solutions.html, October 2024.

[77] SC Magazine. Here's how cybercriminals bypass EDR and why security teams need a defense-in-depth approach. https://www.scmagazine.com/perspective/heres-how-cybercriminals-bypass-edr-and-why-security-teams-need-a-defense-in-depth-approach, June 2023.

[78] Usman Sikander. AV/EDR evasion using direct system calls (user-mode vs kernel-mode). https://osintteam.blog/av-edr-evasion-using-direct-system-calls-user-mode-vs-kernel-mode-fad2fdfed01a, March 2022.

[79] Ankit Sinha. A deep dive into EDR bypass strategies. https://medium.com/@ankitsinha81195_47457/a-deep-dive-into-edr-bypass-strategies-ed25b3929bb1, March 2024.

[80] SOCRadar. The wolf in sheep's clothing: How cybercriminals abuse legitimate software. https://socradar.io/the-wolf-in-sheeps-clothing-how-cybercriminals-abuse-legitimate-software/, August 2023.

[81] Sophos. Results from the 2023 MITRE Engenuity ATT&CK Evaluations (Round 5: Turla). https://news.sophos.com/en-us/2023/09/20/results-from-the-2023-mitre-engenuity-attck-evaluations-round-5-turla/, 2023.

[82] Sophos. Sophos central: Deployment help. https://docs.sophos.com/central/partner/help/en-us/Help/Deployment/index.html, August 2024.

[83] Ryan Stamp. Living-off-the-Land abuse detection using Natural Language Processing and supervised learning. *ArXiv*, abs/2208.12836, 2022.

[84] Symantec Threat Hunter Team. Data exfiltration: Increasing number of tools leveraged by ransomware attackers. https://symantec-enterprise-blogs.security.com/threat-intelligence/ransomware-data-exfiltration, March 2024.

[85] Tanmay Ganacharya (Red Canary). T1055.001 - process injection: Dynamic-link library injection. https://github.com/redcanaryco/atomic-red-team/blob/master/atomics/T1055.001/T1055.001.md, 2023.

[86] Lumu Technologies. EDR evasion: Techniques and strategies to avoid detection. https://lumu.io/blog/edr-evasion/, April 2023.

[87] Trend Micro. Trend Micro achieves 100% protection rate in rigorous MITRE Engenuity ATT&CK evaluations. https://newsroom.trendmicro.com/2023-09-20-Trend-Micro-Achieves-100-Protection-Rate-in-Rigorous-MITRE-Engenuity-ATT-CK-R-Evaluations, 2023.

[88] Dmitrijs Trizna, Luca Demetrio, Battista Biggio, and Fabio Roli. Living-off-The-Land reverse-shell detection by informed data augmentation. *ArXiv*, abs/2402.18329, 2024.

[89] Vaadata. Antivirus and EDR bypass techniques. https://www.vaadata.com/blog/antivirus-and-edr-bypass-techniques/, February 2024.

[90] WebSec. Security software process and driver names. https://github.com/websec/Security-Software-Process-and-Driver-Names, 2024.

[91] Nicholas White. Macros from the internet are blocked by default in office. https://learn.microsoft.com/en-us/microsoft-365-apps/security/internet-macros-blocked, July 2024.

[92] WithSecure. Sending installation link via email. https://www.withsecure.com/userguides/product.html?business/psb-portal/latest/en/task_4BBCD978718B46C2ACD5D21B65A59DDC-psb-portal-latest-en, 2024.

A EDR Drivers and Processes

Table 3: List of EDR solutions with associated, drivers, and processes; grouped by type. This table focuses on core drivers and services installed with the EDR solution, rather than listing all processes associated with the product. – indicates that the EDR solution does not utilize kernel drivers or that specific driver information is not publicly available. We observed some drivers and processes are shared with EPP solutions offered by the same vendor. While we aimed for comprehensive coverage, some EDR solutions may not be included in this table.

Company	EDR Solution	Driver(s)	Process(es)
		Proprietary	
Microsoft	Defender for Endpoint	WdFilter.sys, WdNisDrv.sys, WdBoot.sys	MsSense.exe, SenseIR.exe, SenseNdr.exe, SenseCncProxy.exe, SenseSampleUploader.exe
CrowdStrike	Falcon	im.sys, csagent.sys	csagent.exe, CSFalconService.exe
SentinelOne	Singularity	SentinelMonitor.sys	SentinelAgent.exe, SentinelAgentWorker.exe, SentinelServiceHost.exe
VMware	Carbon Black	carbonblackk.sys, cbk7.sys, cbstream.sys	cb.exe, RepMgr.exe, RepUtils.exe, RepUx.exe, RepWAV.exe, RepWSC.exe
Cisco	Secure Endpoint	csaav.sys, csaam.sys, csacentr.sys, csaenh.sys	sfc.exe
Cybereason	Defense Platform	CRExecPrev.sys	AmSvc.exe, CrAmTray.exe, CrsSvc.exe, ExecutionPreventionSvc.exe, CybereasonAV.exe
BlackBerry	CylanceOPTICS	cyoptics.sys, CyProtectDrv32.sys, CyProtectDrv64.sys	CylanceSvc.exe, CyOptics.exe
Elastic	Elastic EDR	elastic-endpoint-driver.sys, ElasticElam.sys	elastic-endpoint.exe
ESET	Inspect	edevmon.sys, ehdrv.sys, eamonm.sys	EIConnector.exe, ekrn.exe
Palo Alto Networks	Cortex XDR	cyverak.sys, cyvrfsfd.sys, cyvrmtgn.sys, tedrdrv.sys	cyserver.exe, CyveraService.exe, CyvrFsFlt.exe
Trellix	Trellix EDR	FeKern.sys, WFP_MRT.sys	xagt.exe
Fortinet	FortiEDR	FortiEDRBaseDriver_*.sys, FortiEDRElamDriver_*.sys	FortiEDRCollector.exe, FortiEDRCollectorService.exe
HarfangLab	HarfangLab EDR	hlprotect.sys	hurukai.exe
Qualys	Qualys EDR	QMON.sys, qfimdvr.sys	QualysAgent.exe
Trend Micro	Apex One	TmKmSnsr.sys	EndpointBasecamp.exe, WSCommunicator.exe
Sophos	Sophos EDR	SophosED.sys	SEDService.exe, SophosLiveQueryService.exe
Tanium	Tanium EDR	TaniumRecorderDrv.sys	TaniumClient.exe, TaniumCX.exe
Malwarebytes	ThreatDown EDR	mbam.sys, FlightRecorder.sys, MbamChameleon.sys	MBCloudEA.exe, ARSLauncher.exe, EAServiceMonitor.exe
WithSecure	WithSecure EDR	fshs.sys, fsatp.sys	fshoster64.exe, fshoster32.exe
Broadcom	Symantec EDR	fencry.sys, symrg.sys	SemSvc.exe, snac64.exe
Kaspersky	Kaspersky EDR	klif.sys, klhk.sys, klflt.sys	klnagent.exe, avp.exe
LimaCharlie	LimaCharlie EDR	tmp_hbs_acq.sys	rphcp.exe
		Open-Source	
Wazuh	Wazuh EDR	–	ossec-agent.exe
Xcitium	OpenEDR	–	edrsvc.exe
Bluespawn	Bluespawn EDR	–	bluespawn.exe