E

EDR: Endpoint Detection and Response Explained

Learn what EDR collects, how it detects and investigates endpoint attacks, how response actions work, and how EDR differs from antivirus, XDR, and MDR.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
3 min

CONTENTS

In this guide

01 What does EDR stand for?
02 What EDR observes
03 Detection, investigation, and response
04 EDR versus antivirus
05 EDR versus XDR and MDR
06 What EDR cannot do alone
07 Deployment checklist
08 Useful success metrics

Endpoint Detection and Response (EDR) continuously collects selected security telemetry from endpoints such as laptops, desktops, and servers. It analyzes behavior, creates alerts for suspicious activity, gives investigators historical context, and provides response actions such as isolating a device or stopping a malicious process.

01 What does EDR stand for?

EDR stands for **Endpoint Detection and Response**. “Endpoint” means a monitored device such as a workstation or server; “detection” means identifying suspicious behavior from telemetry; and “response” includes actions such as isolating a host, killing a process, quarantining a file, or collecting investigation data. The acronym describes a security capability, not one standardized feature set.

02 What EDR observes

Coverage varies by product and operating system, but EDR telemetry can include:

- Process creation, command lines, parent-child relationships, and code-signing information.
- File creation, modification, hashes, and persistence locations.
- Registry, service, scheduled-task, and startup changes.
- User sign-ins, privilege changes, and interactive sessions.
- Endpoint network connections, DNS activity, and listening services.
- Security-product tampering and selected memory or kernel behavior.

EDR is not necessarily a complete audit archive of every endpoint event. Sensors can filter, aggregate, or throttle data, so retention and collection details must be understood before an investigation.

03 Detection, investigation, and response

1. **Detection:** analytics, behavioral rules, threat intelligence, and custom queries identify suspicious activity.
2. **Investigation:** analysts review the process tree, timeline, affected users, files, connections, and related endpoints.
3. **Response:** authorized actions can isolate a device, kill a process, quarantine a file, collect evidence, block an indicator, or start a scan.

4. **Recovery:** teams remove persistence, rotate credentials, restore trusted state, and monitor for recurrence.

Automation should be scoped by confidence and business impact. Isolating a critical server or deleting a file automatically can cause an outage when context is wrong.

04 EDR versus antivirus

Traditional antivirus focuses heavily on preventing and removing known malicious files. Modern endpoint protection can also use behavior and cloud intelligence. EDR adds investigation history, threat hunting, incident context, and response workflows. These functions commonly operate together rather than as mutually exclusive products.

05 EDR versus XDR and MDR

- **EDR** centers on endpoint telemetry and endpoint response.
- **XDR** correlates endpoint signals with additional domains such as identity, email, cloud applications, and networks.
- **MDR** is a managed service in which external analysts help monitor, investigate, and respond using EDR, XDR, SIEM, or other tools.

06 What EDR cannot do alone

- It cannot protect devices that are unsupported, not onboarded, offline, or running a broken sensor.
- It may lack complete visibility into email, identity, SaaS, unmanaged devices, and network appliances.
- Encrypted or in-memory activity can reduce available evidence.
- Misconfigured exclusions and excessive alert noise can hide genuine attacks.
- An alert still requires triage, ownership, and a tested response process.

07 Deployment checklist

1. Inventory supported endpoints and define ownership for servers, workstations, and remote devices.
2. Pilot sensor compatibility, CPU, memory, storage, network use, and business applications.
3. Protect sensor configuration and administrative roles with least privilege and MFA.
4. Monitor onboarding status, sensor health, policy drift, and stale agents.
5. Tune exclusions narrowly and review them regularly.
6. Define retention and privacy rules for command lines, usernames, file paths, and collected evidence.
7. Test isolation, evidence collection, escalation, and restoration before an incident.

08 Useful success metrics

Measure coverage and response quality rather than raw alert count: healthy sensor percentage, time to triage, time to contain, repeated incidents, false-positive rate, unreviewed high-severity alerts, and the percentage of critical systems with tested response procedures.

NEED MORE HELP?

We are here when the guide is not enough.

Visit the Gridinsoft Support Center for current product guidance or submit a support request with the details of your case.

Open this guide online at support.gridinsoft.com