

EDR evaluation guide

Key considerations for
today's cybersecurity
landscape



How we got here

A brief history of EDR

“

...we continue to detect pervasive threats and prevent impactful intrusions with endpoint detection and response (EDR) tools.”

Concerns over cloud and identity security have dominated the information security space in recent years. While it's important for organizations to properly configure and monitor their cloud and identity infrastructure, we continue to detect pervasive threats and prevent impactful intrusions with endpoint detection and response (EDR) tools. The choice to forgo quality EDR—or the failure to diligently monitor its output—puts an organization at high risk of experiencing a serious security incident.

EDR tools emerged more than a decade ago, addressing a longstanding blindspot among organizations: the need for comprehensive visibility across their endpoint workstations to enable investigation of incidents, perform root cause analysis of alerts generated by a growing array of preventive products and controls, and execute response actions accordingly. Countless tools existed for blocking threats based on known-bad indicators—such as malware signatures and IP addresses—and by other opaque means, but they lacked context and fidelity and offered very little in the way of response actions.

As such, fully investigating and remediating an alert or incident often involved navigating a labyrinth of endpoint logs and searching through staggering volumes of network telemetry based on limited information from dubious alerts, sometimes in a security information and event management (SIEM) platform, if you were fortunate enough to have one. Actually responding to these incidents was another problem altogether. This ranged from physically interacting with hundreds of individual compromised devices to using any variety of native or third-party tools to remotely access them and remediate the incident.

A game changer for investigation and more

The advent of EDR ushered in a new world where data sources that were once scattered across numerous logs (or not collected at all) were now collected by a single tool, storing the corresponding telemetry in a single place, and making it available for analysis in a single search console. Additionally, the “R” of EDR enabled security teams to actively respond to an incident using the same tool they used for investigation, isolating machines and blocking hashes or IP addresses as needed in real time.

Default EDR alerts suffered many of the same problems of alerts in general (high false positive rates, lack of context, too much volume, etc.), but the depth of visibility EDR tools provided was immediately beneficial for retrospective investigations. While early EDR adopters almost exclusively used these tools for retrospective investigation, they had so much more to offer than that.

Sufficiently resourced organizations could also perform ad hoc threat hunts, forward EDR telemetry into a SIEM with other security data, or develop their own detectors that suited their unique business needs and aligned with their specific risk tolerance. Threat hunting and custom detection in particular were places where EDR shined, but dedicated threat hunters and detection engineers were a novelty in the early days of EDR, and most organizations were limited in their capacity to operationalize these functions and truly optimize their EDR investment.

A booming, confusing market

Over the years, EDR has become a near-necessity for all but the smallest organizations. [In a 2023 survey, Gartner found](#) that roughly 57 percent of organizations have EDR capabilities deployed, a 15 percent increase since 2022. Consolidation and evolution have broadened the scope of EDR tools, effectively merging antivirus (AV), next generation antivirus (NGAV), endpoint protection platforms (EPP), and other formerly distinct tools into the

EDR platforms we all recognize today. Further, many modern EDR platforms have rebranded as—or are difficult to differentiate from—extended detection and response (XDR) platforms, which integrate a wide variety of network, identity, and other classes of security tools.

57% of organizations have EDR capabilities deployed

Most of the EDR platforms on the market today boast a core set of features that are fundamentally similar to, but more expansive than their predecessors—they collect telemetry, generate alerts, prevent activity they deem malicious, enable response actions, and present information to security teams in a user interface (UI). However, the specifics of what they collect, prevent, alert on (and under what conditions), the quality and intuitiveness of their response capabilities, and the additional features they offer can vary in dramatic and subtle ways. And these differences matter.

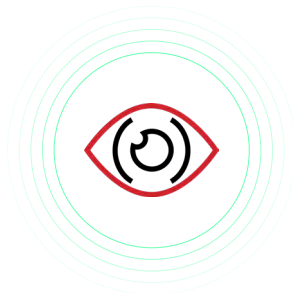
There is no “best EDR platform,” but there is definitely a best EDR platform for your organization. In this guide, we will leverage the experience and expertise we have gained evaluating and operationalizing EDR platforms over the last decade—and guiding thousands of organizations through the process of buying and operationalizing one—to explain what you should consider looking for in an EDR platform, the features you might expect to find (or not), and how they may differ from one EDR product to another. By reading this guide, you’ll be able to gain a better idea of what’s available, use that information to ask the right questions of EDR vendors, and ultimately make the decision that makes the most sense for your unique organization.

“There is no ‘best EDR platform,’ but there is definitely a best EDR platform for your organization.”

About this guide

This guide is a product-agnostic assessment of aggregate features available across all of the EDR platforms we have analyzed and/or integrate into our product. Since the overall quality of an EDR product nearly always depends on the needs and preferences of individual organizations, we will not be recommending or even mentioning specific EDR providers or products in this guide. Instead, we discuss the features we believe are important—and the nuances you might find in the implementation of those features—across the components of EDR listed below.

Since the line between EDR and XDR has become blurry, we also touch on some XDR-like features in this guide because we believe it is important for businesses to consider the benefits they may gain from investing in a tool that correlates endpoint activity with non-endpoint activity. In addition to highlighting EDR features and discussing the ways they might be implemented, we include example questions that you can ask of EDR vendors to learn more about the features they offer and how they're implemented.



Visibility

data sources, reliability of telemetry, and data access



Alerting

volume and fidelity, transparency and criteria, context, and tuning



Prevention

what gets blocked, blocking mechanism, and controls



Response

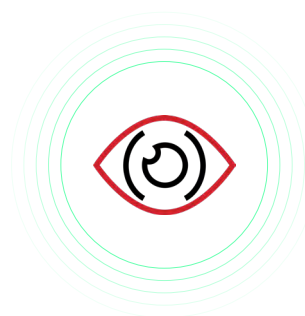
available response actions and protections



Reporting

aggregate alert reports and inventory reporting

Visibility



“If an EDR provider’s data standardization is particularly unusual, you may encounter serious problems unlocking its full potential.”

Visibility is the foundation of an EDR platform, representing the parts of an operating system the tool can collect from. Without visibility, there’s nothing to alert on, prevent, report on, or respond to. We typically categorize visibility into [data sources](#), and MITRE ATT&CK® is the de facto standard for this. ATT&CK further categorizes data sources into components. It is from these data sources and components that an EDR agent collects the actual telemetry that’s monitored for malicious and suspicious events and used to power the EDR platform’s alerting and preventive capabilities.

Data sources

Some examples of data sources, components, and telemetry you might expect an EDR to inspect include the following:

Data Source	Component	Telemetry
Process	Process start	<code>powershell.exe</code>
Command	Command execution	<code>delete shadows /all</code>
File	File access	<code>Q1salesreport.xlsx</code>
Network traffic	Network connection creation	<code>redcanary[.]com</code>
Logon session	Logon session creation	<code>User: Brian</code>

Unfortunately, some EDR vendors use different naming conventions for the kinds of data they collect, but most are conceptually similar and organize loosely into the data sources listed in ATT&CK. The [Open Cybersecurity Schema Framework \(OCSF\)](#) is fast becoming the standard for data standardization and interoperability, so familiarizing yourself with that framework may pay off as vendors continue adopting it. If an EDR provider’s data standardization is particularly unusual, you may encounter serious problems unlocking its full potential.

The following is a list of the kinds of telemetry—organized by ATT&CK data source components where possible—that we consider semi-standard and important for EDR platforms to collect.

General metadata

Event type, endpoint platform, sensor product, sensor ID, and activity timestamp

User information

Username and user UID

Logon session creation and metadata

Account name, domain, logon type, auth type, logon time, access tokens, and success/failure

Process starts

Process ID, process name, process path, process hash, process start timestamp, process command line, and scriptloads

File creations, modifications, and deletions

Filename, file path, and file hash

Network traffic connections

Direction (i.e., inbound or outbound), domain, IP address (remote or local), port, and success/failure

Create, modify, delete registry

Path and value (before and after modification)

Module loads

Module path and name

Process handle open, process thread open, and remote thread creation

Name, path, target, and access requested

“...it’s worth considering the quality of the EDR system’s own logs, given that an EDR platform could inadvertently enable an adversary to manipulate defensive controls or remotely control hosts through response capabilities.”

Beyond these, organizations should also consider whether or not an EDR platform inspects script content via integrations with Microsoft’s [Antimalware Scan Interface \(AMSI\)](#), for example. Having the ability to correlate suspicious activity with corresponding identity telemetry may prove beneficial, as well as information about signing certificates associated with binaries. While an unsigned binary isn’t necessarily malicious in the same way that a signed one isn’t necessarily benign, certificates are an important part of the puzzle when detecting, investigating, and responding to threats. Lastly, it’s worth considering the quality of the EDR system’s own logs, given that an EDR platform could inadvertently enable an adversary to manipulate defensive controls or remotely control hosts through response capabilities.



Reliability

In some cases, the telemetry an EDR agent collects on paper and the actual availability of that telemetry may differ. For example, it may be the case that an EDR platform only presents telemetry that's associated with an alert, severely limiting threat hunting, custom detection, and response use cases. Additionally, some EDR tools implement filtering mechanisms that impact telemetry collection in unpredictable ways. An EDR platform may limit the flow of telemetry under certain conditions or if a type of telemetry is particularly noisy or voluminous, like [registry modifications](#) for example. Some EDR platforms also suffer from process hierarchy issues that can complicate an organization's ability to detect or fully investigate an alert by obscuring what spawned a process or spawned from it. Lastly, it's worth investigating what happens to an endpoint and the telemetry collected from it when it is not connected to the internet.

Data access

Another important component of visibility is how an EDR tool allows its users to access the data it collects. While this section covers graphical interfaces and export functionality, it's important to consider how a person literally accesses the tool as well. The tool should support multi-factor authentication (MFA), role-based access controls, and other identity best practices.

User interface

You'll want to adopt an EDR tool with a user interface that's easy to navigate and intuitive. Some EDRs may present information to you in process trees, timelines, or both. The ideal EDR tool would provide multiple analytical views, including process trees, timelines, user activity logs, and tables,

along with the ability to pivot seamlessly between these, and the ability to sort, filter, and query all the various log types. We discuss this more in the Alerting section.

Search function and query language

Further, each EDR platform has its own unique search functionality and, in all likelihood, query language. Over time, it's possible that product-specific query languages will be replaced by GenAI tools that enable users to forgo learning a proprietary query language and simply search through logs using plain language. Some organizations may prefer to avoid their EDR platform's interface altogether and rely entirely on application programming interfaces (API). Not all EDR platforms offer the same level of API support and some limit API usage, so it's important to evaluate this.

Storage and export capabilities

You may also want the ability to export data out of the EDR tool, so it's worth examining an EDR tool's export capabilities and associated costs. Also, every vendor has its own unique data retention policy for how long they store your endpoint data, and you'll definitely want to know how long the retention period is. Also consider where they store—or can store—the data they collect, both from an operational and data jurisdiction perspective (e.g., on the device, on a local on prem server, in the cloud, in the U.S., in the E.U., etc.).

“Not all EDR platforms offer the same level of API support and some limit API usage, so it's important to evaluate this.”

Questions

What data sources or telemetry types does your EDR tool collect?

What is your EDR tool's naming convention or data standardization model for data sources and telemetry types?

Does your EDR integrate with AMSI?

What XDR integrations does your EDR tool offer?

What kind of logging is available for the EDR tool itself?

In what ways does your EDR tool filter or limit the availability of the telemetry it monitors?

What security protections does your EDR use to prevent unauthorized access?

What query language does your EDR use for search?

What data is accessible via API and are there limits on API usage?

Can you export data from the EDR and is there a cost associated with it?

What is your data retention policy?

Where and how does your EDR store data?



Alerting



Nearly every EDR product includes some level of built-in AV or NGAV features that generate varying amounts of alerts on potentially suspicious and malicious behavior right out of the box. Some organizations will rely heavily on an EDR tool's default alerts while others may ignore them entirely and rely on custom detection rules. Whatever your organization intends to do, you'll have a lot to consider as it relates to the quantity and quality of alerts an EDR tool generates.

Volume and fidelity

Two primary and related concerns for any security tool that generates alerts are the number of alerts it tends to create and the false positive rate of those alerts. Preference for alert volume and fidelity varies from organizations that prefer less volume and higher fidelity—where the risk of missing something might be higher—to organizations that want to see everything even remotely suspicious and are willing to pay the price of excess human investigation of false positive alerts to ensure they don't miss anything. The risk tolerance for most organizations lies somewhere between these two extremes. Attempting to determine the false positive rate for alerts generated by a tool should be a priority for every organization, and the tool should provide a clear means of calculating this.



Transparency and criteria

Understanding what generated an alert and why tells you a lot about the quality of an alert and directs how you respond to it. An opaque alert with no context is extremely difficult to action. Conversely, an alert that explains the logic behind it and includes the specific telemetry that triggered it vastly improves and shortens the investigation process and also unlocks tuning options that might otherwise be impossible.

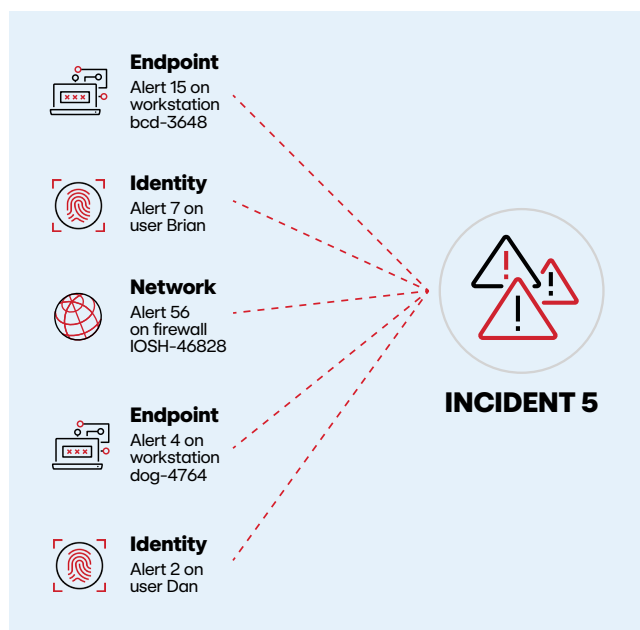
Context is always crucial for an alert, and the context provided by an EDR tool's alerts may differ in subtle but important ways. With alerts generated from **network activity**, knowing whether or not a network connection was successful is important. It may seem obvious that defenders are less concerned about unsuccessful connection or **login attempts** than successful ones, but not all EDR tools make this information easy to determine. Similarly, when an alert relates to a potentially malicious binary, knowing whether or not that binary was signed and having information about the signing certificate is important context.

Alert context

The information provided with an alert varies between different EDR platforms, and the thoroughness of context you receive with an alert is a major contributing factor to the overall quality of an EDR platform—if you care about native alerting at all. The best alerts tell an analyst everything they need to know to **triage, investigate, and respond to a threat**: what happened, who or what was affected, when the activity occurred, why the alert triggered, and how they might respond to it or mitigate the problem that caused it moving forward.

Alert severity is important as well, since it plays a critical role in helping a security team perform triage. Alerts should contain some form of classification to quickly give analysts an idea of the risk posed by the threat that triggered the alert. Mapping alerts to a common standard, like **MITRE ATT&CK**, and associating them with named threats (when possible) can help security teams better conceptualize the intent or purpose of the activity that generated the alert.

If the EDR platform has XDR features, an alert can be automatically correlated with signals or alerts from non-endpoint tooling like an identity provider, email SaaS application, network firewall, and more. Beyond that, having the ability to correlate alerts with one another can be extremely useful as well if, for example, an alert is part of a broader intrusion affecting multiple endpoints or systems. Some EDR platforms aggregate individual alerts into a broader incident, saving security teams valuable time as they triage alerts and try to determine the scope of an incident.



“If the EDR platform has XDR features, an alert can be automatically correlated with signals or alerts from non-endpoint tooling like an identity provider, email SaaS application, network firewall, and more.”

Tuning

Since most organizations have specific expectations and requirements about the kinds of activity for which they want to receive alerts, it's important to understand what level of alert tuning is available for a given EDR platform. The first order of business is gaining an understanding of what's enabled by default and what needs to be manually configured. From there, being able to disable or enable certain alerts or alert types is probably a bare minimum for most organizations (i.e., the ability to disable alerts that generate high volumes of false positives). Assessing the granularity of alert-tuning controls is where things get tricky—and is contingent on the transparency of an alert and the telemetry monitored by the tool.

Having the ability to tune or suppress alerts based on telemetry from the data sources listed in the Visibility section or other criteria can have a significant impact in reducing false positive alerts. For example, you may find that certain users or devices reliably generate a specific alert for activity that you know is benign, in which case it's convenient to tune the alert such that it no longer triggers for activity from that specific user or device. You may find alerts arising when certain processes from permitted tools connect to allowable domains or IP ranges. Being able to tune these alerts to ignore allowable connections but

still trigger for suspect ones is a feature some security teams may want but that not all EDR providers offer.

While it may seem tempting to want an EDR platform that empowers an organization to tune based on every telemetry field it collects, this can also be overwhelming and it's likely that no EDR tool offers tuning on that granular of a level or even provides the alert transparency required to do so. If this is an important use case for your organization, you may be better suited forwarding all telemetry into a SIEM-like tool and developing your own library of custom analytics, at which point you'll want to spend a great deal of time investigating how an EDR tool filters telemetry, the conditions under which it makes telemetry available, and API/data export controls.

“While it may seem tempting to want an EDR platform that empowers an organization to tune based on every telemetry field it collects, this can also be overwhelming...”

Questions

Does your EDR tool include AV/NGAV features?

What is the overall false positive rate of your EDR's alerting?

Does your EDR tool explain the logic it uses to detect threats?

Does your EDR present the telemetry that triggered an alert?

Do alerts include severity, classification, and ATT&CK or threat mapping?

What does an alert look like?

What tuning controls are available for your EDR tool's alerts?

Does your EDR tool aggregate related alerts or events into incidents or otherwise?



Prevention



Note: We discuss security orchestration and response (SOAR) compatibility and features in the Response section, even though they are fundamentally similar to prevention capabilities.

As EDR tools have incorporated more preventive controls, it's become more important to evaluate prevention capabilities as a deciding factor when choosing or assessing an EDR platform. Since an EDR platform's alerting and preventive features can be closely integrated, alerting preferences may have downstream effects on prevention preferences, and therefore organizations should consider how an EDR tool's alerts—and their customization of those alerts—might trigger preventive actions. Importantly, alerts on their own are just noise, whereas prevention is action, and preventive actions can cause real and costly business impacts if you aren't careful. However, there are elements of prevention that should be considered outside of what we've discussed so far.

What gets blocked

Figuring out what kind of activity an EDR tool blocks and when it blocks that activity is an important part of the EDR assessment process. However, preventive controls across many EDR products are variable and dependent on circumstance. As such, it's often hard to get a straightforward or holistic understanding of what any given EDR tool blocks. Even so, organizations should attempt to determine whether and under what conditions an EDR tool blocks the following:

- known malicious malware ([ransomware](#) in particular)
- dual use or offensive security tools (e.g., [Cobalt Strike](#), [Mimikatz](#), [Impacket](#), etc.)
- adversary emulation and breach attack simulation tools (e.g., [Atomic Red Team](#))
- [potentially unwanted programs](#)
- adware
- suspicious or malicious behaviors
- legitimate but often abused software like unapproved [remote management and monitoring \(RMM\) tools](#)
- [LOLbins](#) or native processes
- [files](#)
- [network connections](#)
- [credential theft](#)



Outside of obviously malicious activity, organizations should attempt to learn the criteria an EDR applies when it blocks suspicious activity, particularly for legitimate applications and executables. As with alerting, false-positive block actions are an issue as well but here they can have significant downstream impacts on other parts of the business. Last but certainly not least, it's important to know if the EDR tool blocks before, during, or after execution.

Further, organizations should try to understand the relationship between native alerting and prevention for a given EDR tool, and whether or not they can configure certain kinds of alerts to initiate preventive controls. Equally important is whether the EDR tool can prevent certain kinds of alerts from initiating preventive controls in case their inadvertent use were to cause severe negative business impact.

Blocking mechanism

Understanding how an EDR tool blocks malicious or suspicious activity may be important to some organizations as well. An EDR may block activity via runtime inspection based on signatures or other indicators, by performing static binary analysis, or through some kind of sandboxing. Blocks could also

be based on application policies that are dependent on trusted or untrusted publishers or installers. Or it may leverage some kind of application control mechanism.

Controls

Maybe the most important consideration for an EDR tool's preventive capabilities is the degree to which you are able to configure them. Organizational preference might matter here more than anywhere else. Some organizations want to block first and ask questions later; others have an extremely conservative approach to prevention. Some organizations prefer flexible policy controls that allow them to set prevention conditions on an extremely granular level (e.g., based on file paths). Other organizations might find this level of tuning and configuration overwhelming.

Questions

What does your EDR tool block by default?

How and when does your EDR tool ban indicators of compromise (IOC)?

What is the false positive rate of your EDR tool's preventive controls?

How does your EDR tool block malicious activity?

Does your EDR tool prevent malicious activity before, during, or after execution?

What is the relationship between alerts and prevention in your EDR tool?

Are preventive controls tunable in your EDR tool?

How granular are the preventive tuning controls in your EDR tool?



Response



“Not all EDR products have response parity.”

An EDR tool’s response capabilities are another critical component that organizations should carefully evaluate. You’ll want to determine the response actions the tool is able to execute, the granularity of those response actions, and the security of those response capabilities.

Available response actions

We recommend having the ability to execute the following actions manually, automatically, or preferably both, in direct response to an alert or custom detection analytic:

- ban and unban binaries
- kill processes
- isolate and de-isolate endpoints
- quarantine files
- delete/capture files
- delete registry keys
- block IP addresses, netblocks, ports, and domains
- block users/accounts

Assuming you can set up alert-based triggers to automatically execute the above actions, the response component of your EDR tool effectively serves as a lightweight security orchestration and response (SOAR) tool that can assist security teams quickly containing high-risk threats. It’s also worth considering whether these actions are available via an API. Relatedly, many EDR tools integrate with full featured SOAR tools, via APIs and other means, so organizations that rely extensively on SOAR tools will want to consider this.

However, fully eradicating a threat from an endpoint often requires more than just the actions listed above. Many EDR tools leverage [PowerShell](#) or a custom version of PowerShell for response that offers a robust list of response abilities. However, not all EDR products have response parity. For example, a custom, product-specific version of PowerShell may lack features that are available from an EDR that uses the standard distribution of PowerShell. Any EDR vendor should be able to explain precisely how their response features work and list the actions it can perform.

Response security concerns

It can not be overstated how important it is to make sure these response actions are sufficiently locked down. EDR tools by their core nature have a deep level of access to the endpoints on which they're installed, and response actions are intended to allow someone to remotely perform nearly any action they want on an endpoint. It's bad enough if an adversary somehow manages to gain access to an EDR platform to manipulate preventive or alerting controls to their advantage, but it's far worse yet if they can then use the tool's extensive response capabilities to execute malicious actions across every endpoint protected by the EDR tool. At a bare minimum, MFA should be required by default for all response sessions.

"At a bare minimum, MFA should be required by default for all response sessions."



Questions

What response actions are available in your product?

Is it possible to set triggers that automate certain response actions based on alert criteria?

Are response actions available via an API?

Does your EDR integrate with SOAR products?

How do you secure your response capabilities against compromise?

Is your response functionality protected by MFA?



Reporting



“Tracking these kinds of metrics over time offers organizations a measurable view of their security progress.”

There are two categories of reporting to consider when assessing an EDR product. Long-term reports show aggregate data about alerts and incidents over time, which are directly related to security. The second kind of report itemizes securable assets across your organization and exists at the cross section of IT inventory management and security. It’s also worth looking into whether or not an EDR vendor generates these reports continuously, at scheduled intervals, or only upon manual request.

Aggregate alert reports

Most EDRs tools provide organizations with the ability to generate reports that show aggregate information about alerts. In addition to showing the number of alerts the EDR tool generated, these reports may include other information as well, including:

- the number of alerts by severity level
- the number of alerts by classification
- adversary technique and tactic prevalence (ideally mapped to a standard like ATT&CK)
- named threat prevalence
- preventive actions taken
- response actions implemented

The above list is non-exhaustive, but organizations should consider what kinds of reports they would find valuable and evaluate EDR vendors accordingly. Tracking these kinds of metrics over time offers organizations a measurable view of their security progress. These are especially useful for organizations that perform extensive, repeated testing and use the output of those tests to implement new security controls. Similarly, these reports can also demonstrate whether—and the degree to which—a security team is able to take information from alerts and incidents and use it to improve their security posture.

Inventory reporting

A perhaps less obvious reporting use case is inventory management. Some EDR products provide a snapshot of information about secured or securable assets across an organization. Some examples might include:

- the number of devices and the operating system versions
- the number of identities across an organization
- software installed across an organization and their respective versions
- the number of devices online
- offline devices and the last time they checked in

Again, this list isn't exhaustive, but this is an ideal "off-label" use of an EDR tool, since it already has deep and broad data collected from the systems within the environment. These aggregated insights can inform business decisions beyond those in the security realm, demonstrating extra value for the EDR tool itself.



Questions

What reporting is available in your EDR tool?

Are these reports generated automatically?

What information is contained in these reports?

Does your EDR do any level of inventory reporting, including operating system and software version details?





Conclusion

Beyond the considerations explicitly described in this guide, organizations should consider the usual factors that weigh into all security and IT purchasing decisions, including operating system support, what kinds of support and upgrades the vendor provides during the service period, pricing, regulatory requirements, and more.

About Zscaler

Zscaler (NASDAQ: ZS) accelerates digital transformation so customers can be more agile, efficient, resilient, and secure. The Zscaler Zero Trust Exchange™ platform protects thousands of customers from cyberattacks and data loss by securely connecting users, devices, and applications in any location. Distributed across more than 150 data centers globally, the SSE-based Zero Trust Exchange™ is the world's largest in-line cloud security platform. Learn more at [zscaler.com](https://www.zscaler.com) or follow us on Twitter [@zscaler](https://twitter.com/zscaler).

© 2026 Zscaler, Inc. All rights reserved. Zscaler™ and other trademarks listed at [zscaler.com/legal/trademarks](https://www.zscaler.com/legal/trademarks) are either (i) registered trademarks or service marks or (ii) trademarks or service marks of Zscaler, Inc. in the United States and/or other countries. Any other trademarks are the properties of their respective owners.



**Act Fast.
Stay Secure.™**