YOUR ATTACK SURFACE IS YOUR RELIABILITY PROBLEM

Why Security and SRE need the same current inventory

By: Jeff Collins

Version 1.1 — January 2026

Executive Summary:

Most teams lose time during incidents because they cannot quickly confirm what systems exist, how they connect, and who owns them.

Security and SRE often work from different inventories. That creates extra manual work day to day, reconciling conflicting lists, confirming ownership, and verifying what is actually in use. In most organizations, systems, routes, and third-party dependencies change often enough that inventories and diagrams fall out of date quickly. When speed matters, such as when a serious vulnerability appears, a service slows down, or a provider fails, teams end up rebuilding basic facts under pressure.

The systems that matter most for security are often the same systems that matter most for uptime. This paper explains what a shared inventory needs to include, why it matters during incidents and remediation work, and what to test if you are evaluating tools.

A few situations you may recognize

Here are common situations we hear from teams with strong people and solid tools:

- A serious vulnerability alert arrives and people immediately ask if the affected system is reachable from the internet, whether it is used in production, and who owns it.
- A partial outage hits and the first hour is spent confirming what is actually in the request path and what else depends on it.
- A third-party slowdown causes customer pain and teams manually trace which products rely on that provider.
- Customers report errors but dashboards look fine because the issue sits in a dependency that is not mapped or monitored.
- An audit request arrives and the team spends days rebuilding asset lists, ownership, and evidence from multiple sources.

None of these are process failures. They happen when the environment changes faster than inventories and diagrams are updated.

Why the Security and SRE split keeps breaking down

In older environments, responsibilities felt more separate. Security spent most of its time reducing exposure and keeping systems patched. SRE focused on keeping services fast, stable, and available. That division worked when infrastructure changed slowly and most systems lived inside the company network.

Today, many customer paths run through internet-facing services and edge infrastructure, DNS and routing behavior, certificates, cloud services that scale and move, and external providers such as APIs, SaaS platforms, CDNs, ISPs, and carriers.

Automation also creates and retires resources continuously. When traffic starts flowing through different systems than it did yesterday, both security risk and customer impact change.

As a result, Security and SRE now depend on the same underlying information. They need to know what exists, how systems connect, what is reachable from the internet, what is affected when something fails, and who can take action.

What changes when Security and SRE work from the same inventory?

Most teams are used to separate lists because that is how tools have worked for years. The difference with a shared inventory is practical.

During an incident, teams spend less time confirming what is in the customer path and who owns it. During vulnerability work, they can distinguish between issues that look severe on paper and issues that are actually reachable in their environment. During audits, they can answer ownership and dependency questions with current evidence instead of rebuilding it manually.

Where teams lose time without a shared inventory

Most teams feel the pain of split inventories at the worst possible time, when something is already breaking or an urgent vulnerability shows up. Instead of moving straight to action, teams first have to confirm what is real: what exists, how it connects, and who owns it. These are the most common places that time gets lost.

Unknown systems create both security risk and reliability risk

Some of the most damaging systems are the ones nobody thinks they own. These include old domains that still receive traffic, forgotten API endpoints, load balancers that were never retired, SaaS connectors added by teams that moved on, or cloud resources created during incidents and never cleaned up. Security sees exposure and patch risk. SRE sees an unmonitored component in the customer path. Leadership sees outages or findings that come as a surprise.

Internet-facing exposure changes quickly and affects more than security

Many incidents begin at the edge. A denial-of-service attack can look like an outage. API abuse can look like a capacity issue. Routing changes can look like application latency. CDN

or DNS mistakes can look like an application failure. A third-party outage can appear to be an internal issue until proven otherwise. The fastest fix depends on knowing what is connected and what changed.

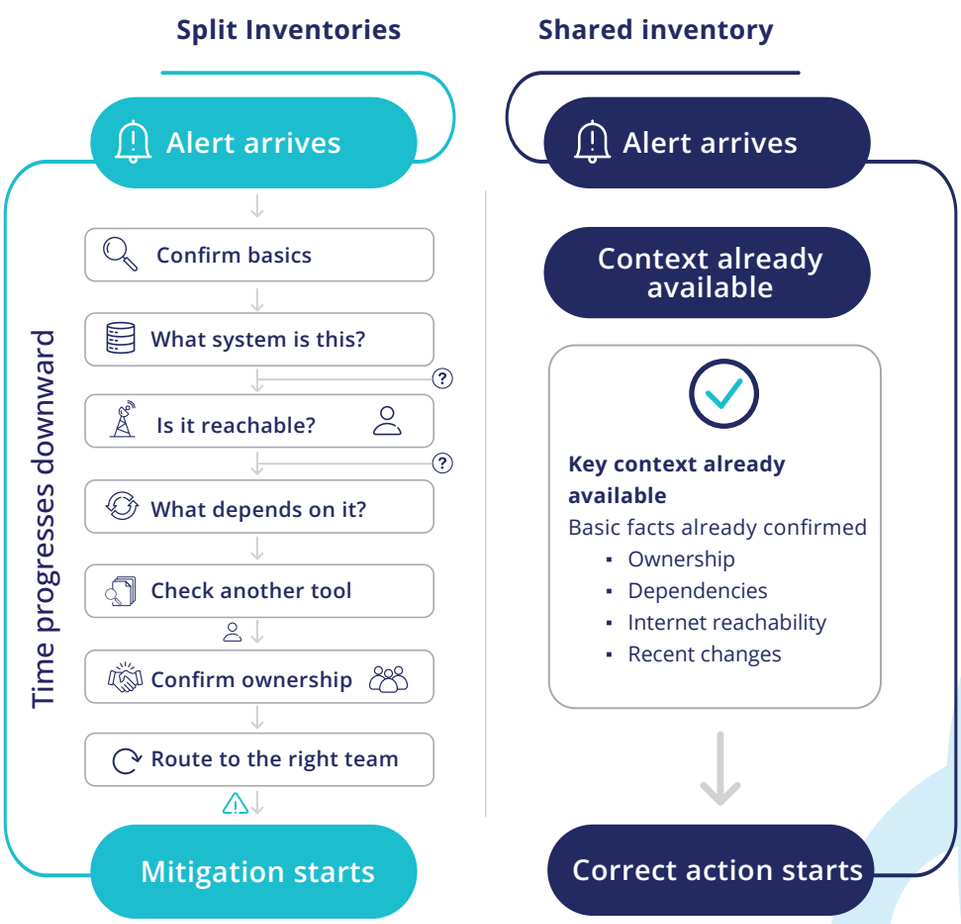
Incident response slows when teams must confirm basics first

During a live incident, teams need fast answers. What is connected to this system right now? What services and customers rely on it? Who owns the systems in the path? What changed recently that could explain this? When those answers live across separate inventories, the first phase of response becomes manual reconciliation. Teams compare lists, validate what is real, and track down ownership.

Compliance evidence requires current operational proof

Audit and risk requests often include practical questions. Who owns this system today? Is it reachable externally? What data does it touch? What depends on it? Can you show evidence rather than assumptions? When inventory data is stale or fragmented, teams rebuild evidence by hand. That work repeats every audit cycle.

How a shared inventory helps teams act sooner



Action starts after basic facts are rebuilt

Correct action begins earlier, with confidence

YOUR ATTACK SURFACE IS YOUR RELIABILITY PROBLEM

What a shared inventory needs to include

A shared inventory helps only if it prevents the same delays during incidents, vulnerability response, and audits.

- **Consistent names and fewer duplicates.** The same service should not appear under different names across tools.
- **Dependency paths you can trace.** Teams should be able to follow what depends on what, including key external providers.
- **Internet reachability.** Teams should be able to tell whether a vulnerable system can be accessed from the public internet.
- **What it can reach next.** If a system is compromised or fails, teams should see what else could be affected downstream.
- **Reliable ownership.** It should be easy to answer who owns the system and who can take action.
- **Recent changes tied to systems.** DNS changes, routing changes, configuration updates, deployments, and certificate updates should be linked to the affected services.

In many organizations, these details exist somewhere, but they are spread across tools, out of date, or hard to confirm quickly.

Common situations where a shared inventory saves time

These examples show what changes when Security and SRE can see the same systems, dependencies, reachability, and ownership.

A third-party API slows down

SRE sees latency and retries. Security may see unusual traffic patterns. A shared inventory helps confirm which services rely on the API, where it sits in the customer path, and which owner needs to contact the provider or reroute traffic.

A forgotten internet-facing system is discovered

Security flags exposure. SRE needs to know whether the system supports a live workflow. A shared inventory helps confirm what traffic reaches it, what it connects to downstream, and who owns the business function it supports.

A VPN gateway misconfiguration causes failures and risk

SRE sees access failures. Security sees unexpected network paths. A shared inventory helps confirm which users and systems rely on the gateway, what paths are open now compared to before, and what changed most recently.

A DNS change causes availability issues

Dashboards may look normal while customers experience failures. A shared inventory helps identify which domains changed, what services and regions are affected, and which teams own the records and routing policies.

A circuit or carrier issue disrupts traffic

The business may not know the circuit exists. SRE may not know it still routes critical traffic. A shared inventory helps confirm where traffic is flowing, which services depend on that path, what alternate routes exist, and who can approve changes.

How teams adopt this in practice

This approach works best when treated as a shared operating decision rather than a single-team project.

Teams start by agreeing on what counts as an asset, including internet-facing services, internal dependencies, external providers, domains, certificates, endpoints, and the systems that carry customer traffic. They map dependencies starting with customer-facing paths and sensitive data systems. Security and reliability signals are tied to the same inventory so exposure findings, performance issues, and change events point to the same system identities. The shared inventory then supports the reviews teams already run, including incident reviews, vulnerability prioritization, change reviews, third-party risk reviews, and audit preparation.

How teams adopt this in practice

This approach works best when treated as a shared operating decision rather than a single-team project.

Teams start by agreeing on what counts as an asset, including internet-facing services, internal dependencies, external providers, domains, certificates, endpoints, and the systems that carry customer traffic. They map dependencies starting with customer-facing paths and sensitive data systems. Security and reliability signals are tied to the same inventory so exposure findings, performance issues, and change events point to the same system identities. The shared inventory then supports the reviews teams already run, including incident reviews, vulnerability prioritization, change reviews, third-party risk reviews, and audit preparation.

What to test in a trial or evaluation

If a tool claims it can support Security and SRE together, these checks show whether it will actually help in real work.

- Can it merge duplicates and resolve naming conflicts across domains, IPs, services, and cloud resources?
- Can it trace a dependency chain from a customer-facing service to the systems and providers it relies on?
- Can it show whether a vulnerable system is reachable from the public internet?
- Can teams quickly identify ownership and escalation paths?
- During a live issue, does it help teams identify what changed recently, what services and customers are affected, who should act first, and what the safest next step is?
- Before a change goes live, can teams see which systems could be affected if something goes wrong?

If a tool cannot pass these checks, it is likely to become another separate inventory.

Where WanAware fits

Some teams address these issues by using a relationship-based inventory that keeps system identity, dependencies, reachability, ownership, and recent changes connected in one place. That structure reduces the time spent confirming basic facts during incidents and helps Security and SRE prioritize work using the same information.

Conclusion

Modern systems change too quickly for fragmented inventories to stay accurate.

When Security and SRE share the same current inventory of systems, dependencies, reachability, ownership, and recent changes, teams spend less time confirming basics and more time fixing the actual issue. Incidents move faster, vulnerability work becomes more targeted, and audit evidence is easier to produce.

If your teams keep rebuilding the same information during high-pressure moments, the most practical next step is to evaluate whether a shared, current inventory can remove that repeated work.

Conclusion

Start Free Trial

See if your teams can confirm what's connected and who owns it, fast.

www.wanaware.com

Version 1.1 — January 2026