Tuskira

PREEMPTIVE EXPOSURE MANAGEMENT

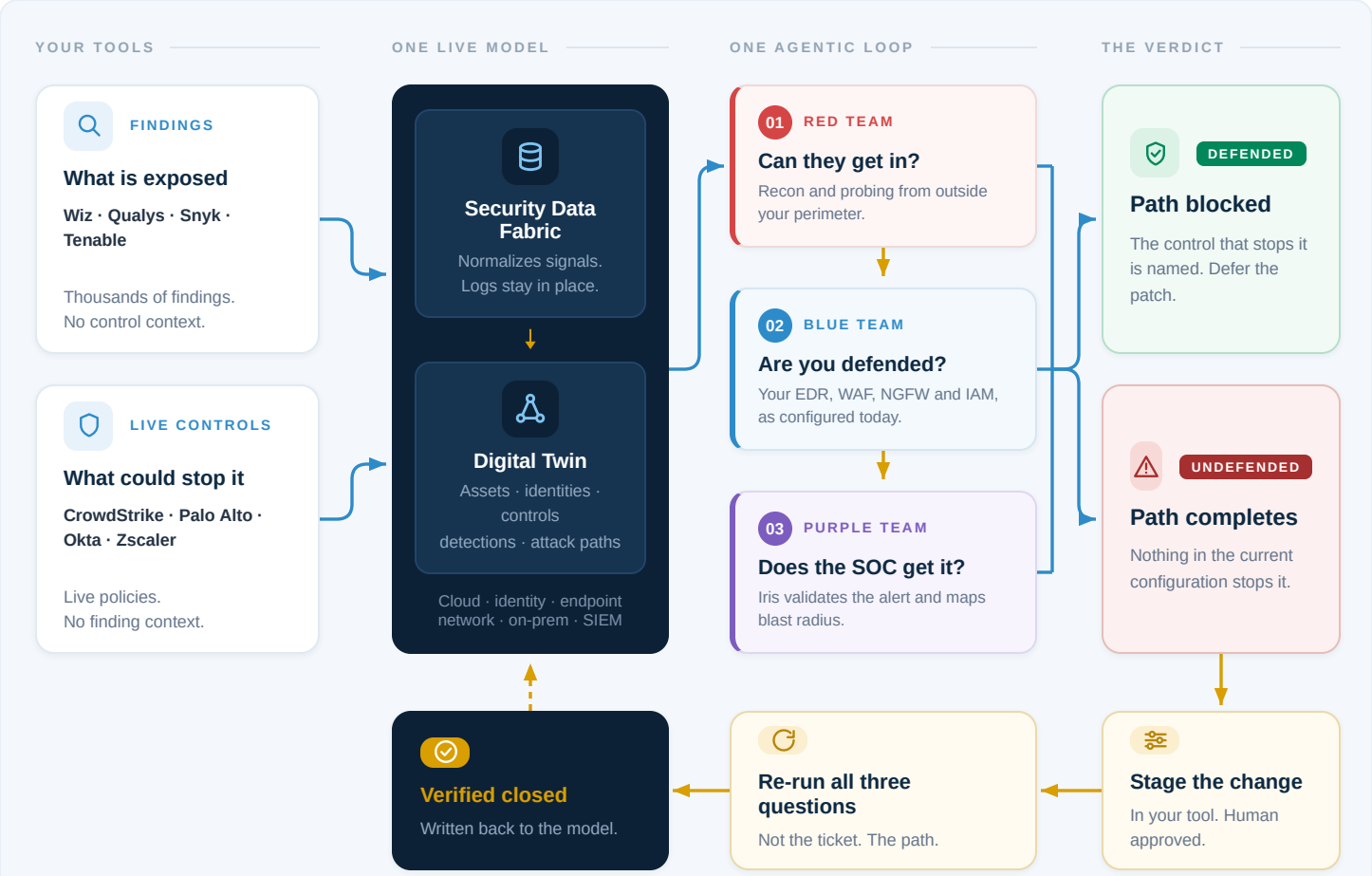
Validate defenses, exposures and breach paths

A scanner reports the exposure. Tuskira proves the path is reachable, tests it against **the controls running today**, identifies the change that breaks it, and re-tests to prove closure.

WHY NOW Frontier AI models like Mythos now chain low-severity findings into working exploits in minutes. Tuskira runs that chain first, against the controls you already have.

THE ARCHITECTURE

Exposed doesn't mean *exploitable.*



ILLUSTRATIVE VALIDATION RECEIPT

```
EVIDENCE  asset conf-dc-02 · finding CVE-2023-22527 (CVSS 10.0) · reachable yes, from edge-prod
 VERDICT  control WAF policy edge-prod-01 · result PATH BLOCKED · re-tested 14:22 UTC · priority defer
```

The point of testing against controls is to spend the team's time only on what is real. Tuskira has deprioritized **up to 99% of scanner findings as not needing action**, whichever tool surfaced them. The reachable findings that remain are then tested against deployed controls and closed.

Governed by design. Validation is autonomous. Containment needs approval, and it runs through the controls you already own.

99%

of scanner findings deprioritized as not reachable, not undefended, or not on a breach path.
Source: Tuskira, Kairo launch.

Tuskira

WHY THE ANSWER HOLDS UP

Third-party controls, not one stack

Your EDR, WAF, firewall and identity provider came from different vendors. Tuskira tests against all of them, so **the verdict never depends on a vendor assessing itself.**

Every domain, not just cloud

A path that starts in cloud and ends in Active Directory is **one path, not two reports.** Cloud, identity, endpoint, network, on-prem and SIEM in one model.

Breach validation, not internet reachability

Internet-facing is not the same as breachable. Tuskira walks the path to a system that matters and **proves whether it can be completed.**

Control assessment, continuously

Each control is assessed for whether it is deployed, enforced and configured to stop what it should. **Gaps surface before an attacker finds them.**

WHAT IT READS

CONTROLS

EDR, WAF, NGFW, IAM, segmentation

CrowdStrike, Palo Alto, Okta, Zscaler and equivalents, read live.

EXPOSURE SOURCES

VM, CNAPP, SCA, SAST, EASM

Wiz, Qualys, Snyk, Tenable, GitHub. One more signal, not a system of record.

DETECTION AND LOGS

SIEM, mXDR, log stores

Splunk, Rapid7, Google SecOps. Queried in place, never centralized.

THREAT INTEL

KEV, EPSS, exploit feeds

Current TTPs and zero-day disclosures, so reachability reflects now.

150+ tools through APIs and MCP. Read-only access, nothing installed on an endpoint. Logs remain in place and are queried through federated search. Relevant security context is normalized into the model.

RED, BLUE AND PURPLE, IN ONE LOOP

Not isolated agents producing signals in silos.

RED TEAM · VECTOR, KAIRO

Test it like an attacker

Recon the external surface, then walk the breach path across cloud, identity, endpoint and on-prem.

BLUE TEAM · LATTICE, QUELL

Prove the defenses work

Assess whether the controls you deployed actually break that path in their current configuration.

PURPLE TEAM · IRIS

Carry it to the SOC

Validate the alert the path produced, map its blast radius and orchestrate the response.

Each agent writes its result into the same model, so the next one starts from it. **The reconciliation a purple team does in a room happens continuously, in the data.**

HOW IT DIFFERS FROM WHAT TESTS THINGS TODAY

THE APPROACH	WHAT IT PROVES	WHERE IT TYPICALLY STOPS
Vulnerability scanner	Something on this asset is vulnerable	Cannot read the configuration of the control sitting in front of it
Attack surface management	An asset is reachable from the internet	Stops at reachability, so it cannot say whether a breach path completes
BAS	A predefined technique executes against an individual control	Often validates set techniques without the cross-domain asset, identity and business context of your path
A control vendor's own validation	Its own telemetry saw the event	Cannot see the other vendors that make up the rest of your estate
Tuskira	**Your specific path, against your deployed controls as configured today**	**Re-tests the same path after the change to prove it stayed shut**

HOW A DEPLOYMENT STARTS

1 Read-only API keys for the tools you already run, nothing installed on an endpoint. 2 The fabric normalizes both sides of your stack into one live model. 3 The first verdict names what is reachable and undefended, with evidence, before anything changes.

Most platforms prove a path is reachable.

Tuskira proves whether the controls you already own stop it, closes what is left through those same tools, and re-tests to confirm it stayed shut.

REQUEST A DEMO

tuskira.ai/request-a-demo

We will run control validation against your own environment and show you what your controls already stop.

© 2026 Tuskira, Inc. · tuskira.ai