

SECURE YOUR NETWORK

Autonomous Security Operations

Architecture Overview

A self-learning defense platform that detects and contains attacks on Microsoft Azure networks automatically.

Product: Secure your Network

Document: Architecture & Design, v0.1 (pre-release)

Date: October 2026

Contact: info@securecompany.ca

1. Executive summary

Modern intrusions move at machine speed. Recorded attacker “breakout” times — the interval between first access and reaching a second system — are now measured in seconds, faster than any human analyst can respond. Signature-based tools miss attacks that adapt and imitate legitimate traffic, and conventional automation relies on brittle hand-written rules.

Secure your Network is an autonomous security operations platform. It builds a live model of a network, continuously simulates attacks against it, and uses a self-learning policy to decide which defensive action contains a threat with the least disruption — then executes low-impact actions automatically and escalates high-impact ones to an analyst. It is built to run natively on Microsoft Azure and augments, rather than replaces, existing Azure security controls.

Status: the core decision engine is validated in simulation (see Section 8). A production system is a staged build; the roadmap is in Section 11.

2. The problem

- **Speed.** compressing the response window below what human-in-the-loop operations can meet.
- **Adaptation.** the attacker reasons over what it finds and reroutes, so fixed defensive rules fail.
- **Evasion.** activity imitates legitimate traffic and authorized actions, defeating signature and pattern matching.

The result is a widening gap between the speed and sophistication of attacks and the speed and precision of defense. Closing that gap requires detection based on behavior rather than signatures, and response that is both automated and safe.

3. Solution overview

The platform operates as a closed control loop with five stages: sense the network, model it, simulate attacks against it, decide on a response, and act. Outcomes feed back to improve the decision policy — and that improvement happens entirely in simulation, never by experimenting on a live network.

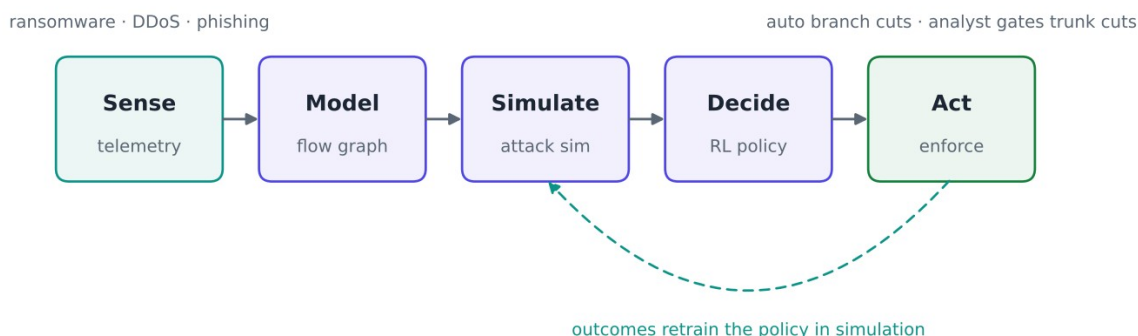


Figure 1. The autonomous security operations control loop.

4. Core components

4.1 Sensing

The platform ingests telemetry already available in an Azure environment: network flow logs, endpoint and identity signals, and security alerts. Different threats draw on different signals (Section 6).

4.2 Network flow-topology model

A proprietary flow-topology model represents the network as a weighted graph and identifies the paths through which most traffic — and therefore most risk — is concentrated, along with the least-resistance route to each protected asset. This model is the platform's core intellectual property; its internal method is a trade secret and is not described in this document. Functionally, it provides two things: a compact, behavior-based representation of the network's structure, and a change signal — an intrusion reshapes the graph (for example, new lateral paths between hosts that never previously communicated) before it does anything a signature would catch.

4.3 Attack simulation

A Monte Carlo engine runs large numbers of simulated attacks against the current model, estimating how likely an intrusion is to reach each protected asset and how much that likelihood changes under any candidate defensive action. The simulation serves two roles: a live risk estimate, and the training environment for the decision policy.

4.4 Decision engine

A reinforcement-learning policy observes the modeled state and selects an action — block a link, rate-limit, isolate a host, or take no action — to minimize risk while minimizing collateral impact on legitimate traffic. The policy is trained against the attack simulation, so it learns effective, non-obvious responses without ever experimenting on production. In validation it independently discovered that the intuitive action (cutting the busiest link) is counterproductive, and learned a markedly better strategy (Section 8).

4.5 Enforcement

Decisions are executed through native Azure controls — network security groups, Azure Firewall, and web application firewall rules — and through identity and endpoint isolation. Every action is reversible and time-boxed.

5. Threat coverage

One decision engine serves multiple threats, but each threat draws on its own detection signals. The platform is strongest where an attack propagates through the network; phishing is detected primarily from email and identity signals, with the network model contributing to post-click containment.

Threat	How it is detected	Automated response
Ransomware	New lateral paths in the flow model, combined with endpoint signals	Isolate the host, disable the account, recover from immutable backups
DDoS	Volumetric floods (handled by Azure DDoS Protection) plus trunk-saturation signals	Application-layer rate limiting; scale or shed load
Phishing	Email and identity signals (sign-in risk, mail security) — not the flow model	Block sender, revoke sessions, reset credentials, contain any clicked-through host

6. Autonomy and safety guardrails

Automated blocking carries real operational risk: a wrong action can cause a self-inflicted outage. The platform is designed so that autonomy is bounded by guardrails that make it safe to deploy.

- **Tiered autonomy:** low-impact actions (narrow blocks, rate limits) execute automatically; high-impact actions (cutting a major path, disabling accounts) require analyst approval.
- **Learn in simulation, act in production:** the decision policy is only ever trained against the simulation. It never experiments on production.
- **Reversible, time-boxed actions:** every enforcement action expires automatically and can be rolled back with one action.
- **Shadow mode first:** on first deployment the platform recommends actions without executing them, until its recommendations have earned operator trust and the simulation has been calibrated to the live environment.

7. Proof of concept and validation

The decision engine has been validated in simulation on a representative network. Trained only against simulated attacks, the policy learned to contain an intrusion — reducing the modeled probability of an attacker reaching the protected asset from roughly 50% to under 1% — while affecting only about 11% of legitimate traffic. It outperformed every hand-written rule, and in particular avoided the common but counterproductive move of blocking the highest-traffic link, which in this scenario increases risk as attackers reroute.

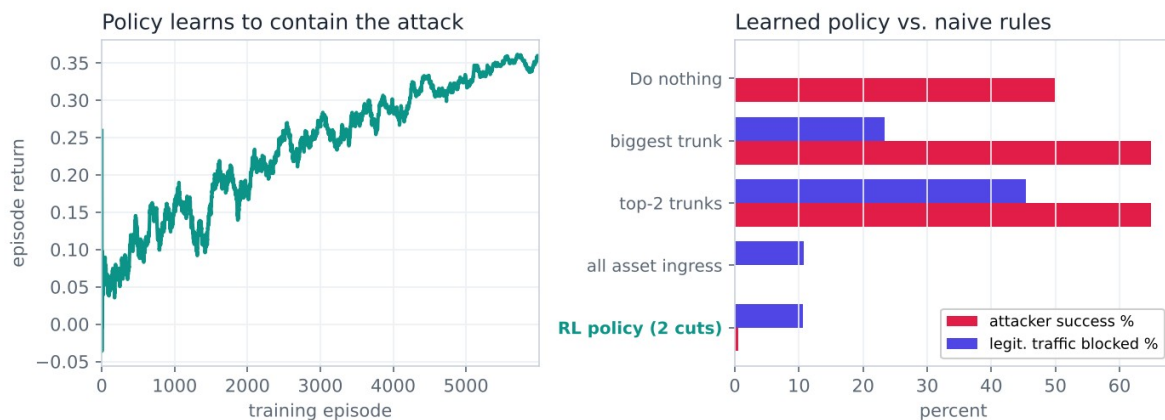


Figure 2. Left: the policy improves over training. Right: the learned policy contains the attack at far lower collateral cost than naive rules.

Strategy	Attacker success	Legitimate traffic blocked
Do nothing	50%	0%
Block the busiest link	65% (worse)	23%
Learned policy	under 1%	about 11%

These figures are from a simulated model, not a production deployment; they demonstrate the decision engine, not a field result.

8. Azure-native integration

The platform is additive to an existing Azure investment. It reads topology and traffic through standard Azure interfaces, authenticates with managed identities (no stored secrets), and enforces through the customer's own network and identity controls. It requires read access to telemetry and scoped permissions to enforce; it installs no agents on production hosts for its core sensing.

9. Security and data handling

- **Least privilege:** read-only for sensing; scoped, least-privilege permissions for enforcement; no standing secrets in code.
- **Data minimization:** the platform operates on network metadata and security telemetry, not payload content.
- **Auditability:** every automated and analyst action is logged and attributable, supporting audit and incident review.
- **Fail-safe defaults:** the system is designed to fail safe — on uncertainty it recommends rather than acts.

10. Roadmap

Phase	Focus	Outcome
1	Decision engine (complete)	Validated self-learning policy in simulation
2	Live integration, shadow mode	Reads a real Azure environment; recommends without enforcing
3	Scale and depth	Larger networks; deeper learning algorithms; per-threat detectors
4	Supervised enforcement	Tiered automated response with analyst approval for high-impact actions
5	Full autonomy within guardrails	Continuous, calibrated, auditable autonomous operation

11. Risks and limitations

- **Simulation-to-reality gap:** a policy optimal in simulation can be wrong in reality; shadow mode and calibration mitigate this.
- **Scale:** the current engine is validated on a small network; production scale requires deeper learning methods.
- **Detection breadth:** phishing and identity threats require detectors outside the network model.
- **Operational risk:** automated enforcement must be introduced gradually, behind guardrails, to avoid operational disruption.