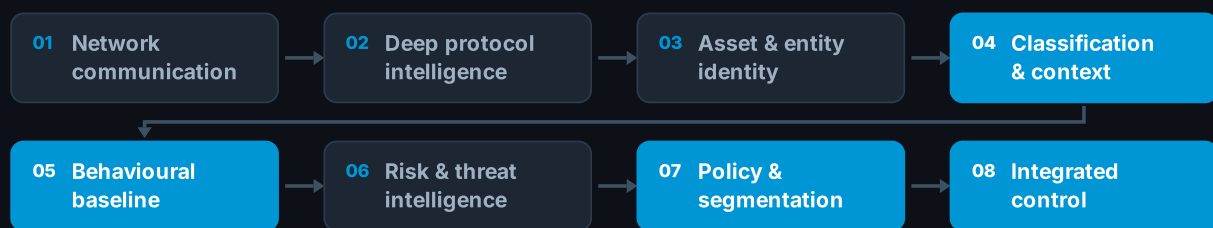


SOLUTION BRIEF

Policy Automation & Micro-Segmentation

Segmentation built on what your network actually does. WireTrace provides the intelligence, policy intent and continuous validation layer, while your enforcement technologies (firewalls, network access control, switches) execute the network control.



For: Security architects, network security and OT security teams

Segmentation projects rarely fail because a firewall cannot filter traffic. They fail because nobody can say with confidence which devices exist, what they are, and which of their conversations production depends on. **WireTrace answers those questions from the network itself, turns the answers into segmentation intent, and keeps checking that intent against reality.**

10

lifecycle steps, from discovery to continuous validation

IEC 62443

zones, conduits and target security levels declared as intent

4

policy run modes, from monitor only to automatic

Append-only

activity log of every policy decision and action

01 Why segmentation stalls

Most organisations already own capable enforcement technology. Firewalls, network access control and managed switches can all restrict traffic precisely. What they cannot tell you is which rule to write. Without that knowledge, teams either write rules so broad they protect little, or rules so tight they stop a process nobody knew existed, and the project is rolled back.

Unknown devices

A cell or ward contains controllers, gateways, panels and appliances that never appear in an asset register, so they never appear in a rule either.

Unknown dependencies

Historians, vendor support paths, time servers, licence servers and engineering tools create conversations that only surface when they are cut.

Unverified intent

Rules are written on paper, applied once and rarely checked again. Drift after maintenance or change goes unnoticed until an audit or an incident.

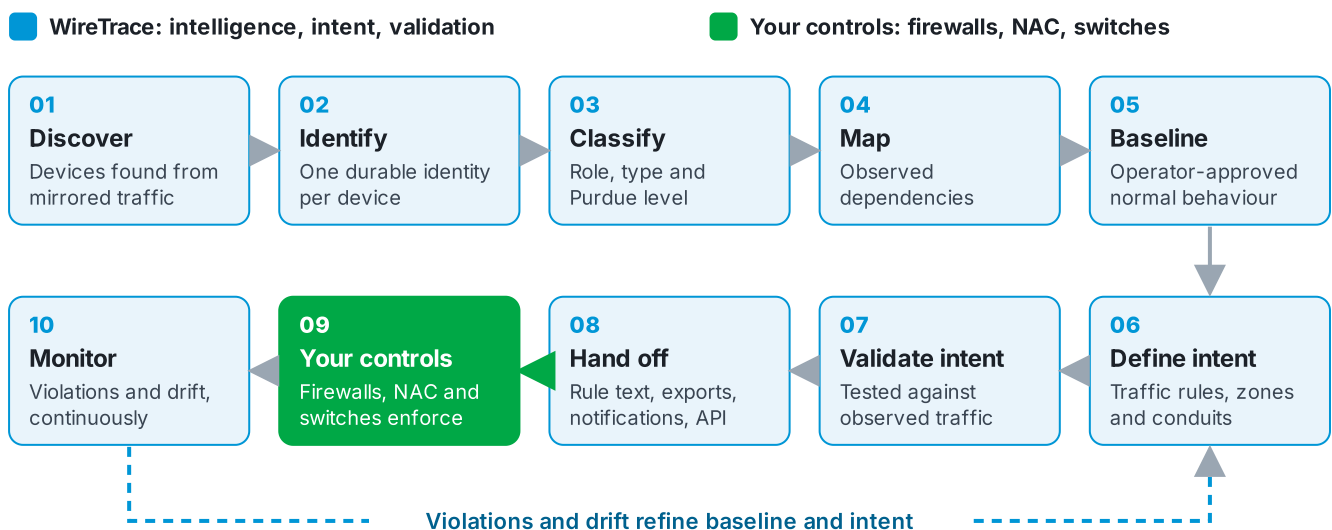
02 A clear division of labour

WireTrace provides the intelligence, policy intent and continuous validation layer, while your enforcement technologies (firewalls, network access control, switches) execute the network control.

This split is deliberate, and it is a strength. The intelligence layer is vendor-neutral: it describes devices, dependencies and intent in terms of the network, not in the syntax of one product, so it stays useful when enforcement technology changes or differs between sites. Your enforcement layer stays under the control of the teams who already operate it, with their own change processes, redundancy designs and audit trails. WireTrace does not push configuration to network devices. It gives those teams the evidence, the validated proposal and the continuous check that make each change safe to approve.

03 The segmentation lifecycle

Ten steps take you from an unknown network to segmentation that is defined from evidence, executed by your controls and validated continuously. Nine of them are WireTrace's job. One, the enforcement itself, belongs to the controls you already own.



Hand-off (step 8) is the boundary: WireTrace proposes and validates, your enforcement layer executes.

Figure 1. The segmentation lifecycle. Steps 1 to 8 and 10 are performed by WireTrace; step 9 is executed by your existing enforcement technologies.

Step	WireTrace	Your enforcement layer
1 Discover	Builds the inventory from mirrored traffic (SPAN, TAP or ERSPAN); optional read-only switch polling adds port-level connectivity.	Provides the mirror or TAP.
2 Identify	Keeps one durable identity per physical device across address changes, interfaces and sensors, with the source of every value.	None.
3 Classify	Assigns domain, device type and role, and a Purdue level with stated confidence and a one-line reason.	None.
4 Map dependencies	Shows observed communication between devices, groups and zones, by protocol and port.	None.
5 Baseline	Records a Communication Baseline and a Protocol Behaviour Baseline for your operators to review and approve.	None.
6 Define intent	Holds your traffic rules (which zones and devices may communicate, in which direction, over which services) and your IEC 62443 zones, conduits and target security levels.	Network and OT owners agree the intent.
7 Validate intent	Tests proposed rules against observed traffic over a bounded window before you rely on them.	Reviews what would have been affected.
8 Hand off	Exports proposed rules as text, publishes blocklists for firewalls to pull, sends notifications and serves context through the REST API.	Receives the proposal in its own change process.
9 Enforce	None. WireTrace does not change device configuration.	Firewalls, NAC and switches execute the control.
10 Monitor	Detects violations and drift against the approved baseline and declared intent, and raises them with evidence.	Acts on violations through its own process.

04 What happens at each stage

Understand the estate (steps 1 to 4)

Sensors receive mirrored traffic and never transmit onto the monitored network. Each device gets one durable identity, a classification by domain, type and role, and a Purdue level with stated confidence; a device the evidence cannot classify stays unknown rather than guessed. The communication map then shows who talks to whom, by protocol and port, between devices, groups and zones: the dependency picture segmentation designs usually lack.

Agree what normal looks like, and what should be allowed (steps 5 to 7)

Communication Baseline. You record a baseline window of observed traffic, review the communication pairs found, and approve, monitor or deny each one. Approved patterns can become traffic rules, so rules can be seeded from a recorded baseline rather than typed from scratch (seeding works at the level of coarse zones; finer rules are refined by your team).

Protocol Behaviour Baseline. For industrial protocols, WireTrace learns which fields and values each protocol normally carries. Once you approve the baseline, a new field or value, such as a previously unseen function code, raises a deviation. Learning is operator-driven: nothing becomes "normal" until someone approves it.

Intent. Traffic rules declare which zones and devices may communicate, in which direction and over which services. Alongside them you declare IEC 62443 zones, conduits and target security levels, and WireTrace shows observed security-level gaps per zone against those targets; levels that cannot be observed from the network are marked as requiring declaration.

Validation. Before anyone relies on a rule, WireTrace tests it against observed traffic over a bounded window and shows which real conversations it would have allowed or flagged. This is where the stopped-production surprise is caught on paper instead of on the plant floor.

Hand off and enforce (steps 8 and 9)

WireTrace hands the validated outcome to the teams and tools that execute control:

- **Rule proposals:** proposed block rules exported as text for iptables, nftables, Cisco IOS and pf, for your firewall team to review and adapt into their own rulebase.
- **Blocklists:** IP, domain and URL blocklists for firewalls and security tools to pull.
- **Notifications:** violations and decisions over syslog, webhook, email and SNMP trap.
- **REST API:** a NAC or other tool can retrieve WireTrace asset identity, classification and risk to enrich its own policy decisions.

Your firewalls, NAC and switches then execute the control, under the change process your organisation already trusts.

Keep checking (step 10)

WireTrace continuously detects traffic that breaks your intent, including IT-to-OT conduit crossings and large Purdue-level jumps, and raises each violation with supporting evidence. Communication drift against the approved baseline is detected on the schedule you set, so after maintenance or change you see whether the segmentation you designed is still the segmentation you have.

05 Policy automation, under control

Policies let you decide what WireTrace does when conditions you define become true, from simply recording the event to acting automatically. They are designed for environments where an unexpected action can matter as much as a missed one.

Conditions

Asset attributes, groups, sites and zones; risk; vulnerabilities, including known-exploited ones; alerts; protocol and communication behaviour; and baseline deviations. A condition that cannot be evaluated is treated as unknown and does not fire.

Scope and schedule

Each policy applies to an asset, group, site or zone. Schedules set effective dates and active hours, and maintenance windows keep planned work from triggering unplanned responses.

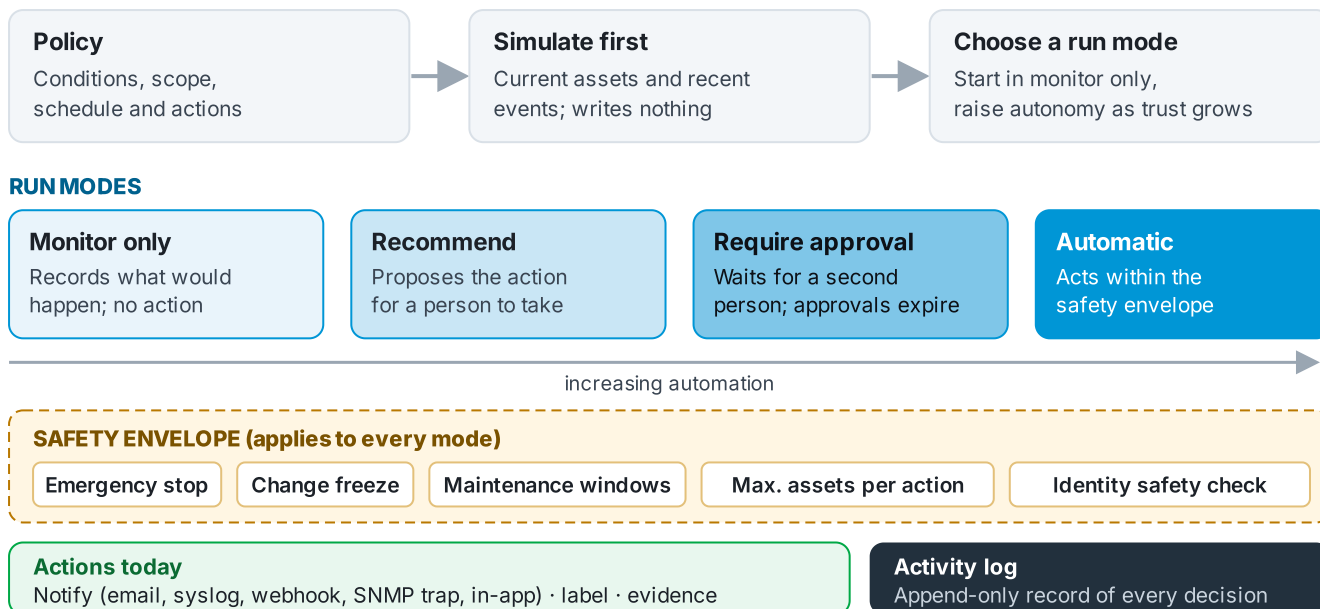


Figure 2. Policies are simulated first, run in one of four modes, and always operate inside the same safety envelope. Every decision and action is recorded in the activity log.

Run modes

Each policy runs as **Monitor only**, **Recommend**, **Require approval** or **Automatic**. Teams typically start every new policy in monitor only, review what it would have done, and raise its autonomy once they trust it.

Approvals and safety controls

- **Four-eyes approvals that expire.** An action that needs approval cannot be approved by the person who requested it, and an approval left unanswered lapses rather than waiting indefinitely.
- **Emergency stop and change freeze** halt automation across the platform when the situation calls for it.
- **Maintenance windows** and a **maximum number of assets per action** keep the reach of any single action bounded.
- **Automation safety check on identity.** Before any automated action, WireTrace checks that the target is identified well enough, and recently enough, to act on safely.

Simulation

Simulate a policy against current assets and recent events before enabling it, and see which assets it would have matched and what it would have done. Nothing is written. Simulation evaluates the present state and recent events; it does not replay historical traffic.

Actions available today, and the record they leave

Policies can send notifications by email, syslog, webhook, SNMP trap and in-app; apply labels to assets; and record compliance evidence. Changes WireTrace makes, such as a label, can be time-bound and undone with a preview. Every decision and action is recorded in an append-only activity log, so the question "who allowed this, and when" always has an answer.

06 In practice

Protect a PLC cell

An OT team wants to isolate a packaging cell. WireTrace discovers its controllers, HMIs and drives, and shows that besides the HMIs, the cell talks to a historian, a time server and an engineering workstation used during maintenance. The team records a baseline, approves the expected pairs, and declares the cell as an IEC 62443 zone with a conduit to supervisory level. The proposed rules are validated against observed traffic, which reveals a vendor support path the design had missed, before it is cut. The customer's firewall team receives the validated traffic rules, together with proposed block rules exported as text for the paths to close, and translates them into its own rulebase through its own change process. WireTrace then watches the conduit for violations.

Brownfield, phased segmentation

A site with decades of flat networking cannot segment in one step. Phase one runs WireTrace in monitor only: inventory, classification, the communication map and baselines, with no rules at all. Phase two declares intent zone by zone and validates it against observed traffic. Phase three hands validated proposals to the network team, whose controls enforce them one zone at a time while WireTrace reports violations and drift for each zone as it goes live.

Drift after change

A shutdown brings contractor laptops, a replaced drive and a temporary remote-access path. Scheduled drift checks compare communication against the approved baseline, and the Protocol Behaviour Baseline flags a function code the cell has never used. The team sees exactly what changed, decides which changes to approve into the baseline, and asks the network team to close the rest.

Unknown device on a controlled zone

A policy scoped to a production zone fires when a device with no approved role appears. In require-approval mode it notifies the OT on-call engineer by email and syslog, proposes an "unverified" label, and waits for a second person to confirm. The label is time-bound; the engineer investigates with the evidence WireTrace recorded, and any network action is taken by the team that owns the switch or NAC.

07 Evidence for audits

Segmentation controls in the compliance frameworks WireTrace maps (including IEC 62443-3-3, NERC CIP and NCA OTCC) are evidenced from observed traffic, and compliance evidence can be exported as a signed evidence bundle. WireTrace provides evidence; auditors and your own assessment close the controls that cannot be observed from the network.

Limits, stated plainly

- WireTrace defines, validates and monitors segmentation intent. It does not push rules or configuration to firewalls, NAC or switches; your enforcement layer executes the control.
- Dependency mapping shows observed communication by protocol and port. It does not establish which side initiated a conversation, and it has no concept of application-level dependencies.
- Rule seeding from a baseline works at coarse zone level. Validation tests rules against observed traffic over a bounded window, and does not simulate a default-deny posture.
- Drift detection runs on the schedule you set and compares observed communication with the approved baseline. WireTrace does not read back firewall state.
- Policy simulation evaluates current assets and recent events; it does not replay historical traffic. Policy actions today are notifications, asset labels and compliance evidence records.
- Coverage follows sensor placement: conversations that never reach a sensor cannot be mapped or validated.

Map one cell before you segment it

Start passively on a mirrored port in front of one production cell or clinical zone. WireTrace shows what is there and who depends on whom, so your first segmentation rule is written from evidence.

Book a demo at wiretrace.io/request-demo or email sales@wiretrace.io