KRY10

Network Guard

Assured Protection for Legacy In-Service and Certified Operational Technology

COMMERCIAL
IN CONFIDENCE



WHITEPAPER_ JUNE 2026

Security isn't a feature, it's *a foundation*



About Kry10

At Kry10, we strengthen the foundations beneath the critical systems that society depends on every day. We bring mathematical assurance to critical embedded and agentic systems, building resilience deep into the technology, right from the foundation.

We work with those responsible for critical systems. From the operators to architects to strategic leaders working across critical national infrastructure, healthcare, defense, telecommunications, transportation and more. All the people who can't afford to get it wrong.

Our focused and growing team across the USA, UK, Australia and New Zealand strengthens what matters for the critical systems society depends on every day.

Our foundations are mathematically proven. Because critical should mean certain.

©2026 KRY10
All rights reserved. Kry10 and associated names, logos, graphics, and product identifiers are trademarks or registered trademarks of Kry10. Any other trademarks referenced remain the property of their respective owners.



Executive summary

Operational technology in critical infrastructure environments is increasingly exposed to network-borne threats it was never designed to withstand. The constraint is structural: many devices cannot be patched without triggering recertification, locking known vulnerabilities in for the operational life of the equipment. The only available mitigations are network-layer controls external to the device.

Kry10 addresses this challenge with the **Kry10 Network Guard**, a physical inline appliance installed between an untrusted network and a protected OT device. It requires no modification to the protected device and no changes to its configuration or firmware. Every packet passes through three inspection phases: rules-based ingress validation, formally verified protocol conformance, and operational policy compliance. Packets that are not permitted, not conformant, or not compliant are detected and, if active filtering is commissioned, dropped before they reach the protected device. For asset owners, the practical effect is that Network Guard extends the operational life of in-service certified equipment by adding a layer of protection without changing the equipment itself.

Network Guard is built on the Kry10 Operating System (KOS), backed by the seL4® microkernel, the only operating system kernel whose isolation and security properties are formally proven. Any compromise of an internal component remains isolated; the boundaries are enforced by a proven kernel.

Network Guard's parsers are generated using EverParse, producing machine-checked proofs of correctness against protocol specifications. Because the specification is fixed, the proof is durable: it does not expire as threats evolve. The KOS platform ensures new parsers for new protocols can be deployed as requirements change, without altering the formally verified foundation beneath them.

The formally verified kernel and formally verified parsers are mutually reinforcing: the kernel ensures correct isolation of Network Guard's internal components, and the parsers enforce tightly controlled communication, ensuring that a malformed packet cannot reach the device. Together they close the attack surface that conventional appliances leave open.



Fighting powerlessness *with agency*

01 Introduction

OT environments have a defining characteristic: the cost of changing the protected asset is dominated by validation, not by the change itself.



A safety-rated controller or certified field device represents years of design review, integration testing, and operational evidence. Modifying it invalidates that history. The constraint applies equally to product certification of the device and to operational accreditation of the system it operates within. Network Guard is designed so that protection does not require OT modification and from that constraint, the rest of the architecture follows. Network Guard's deployment changes the system accreditation context but does not retrigger product certification of the protected device.

Conventional OT firewalls inherit the security model of their underlying general-purpose operating systems, where a single privileged compromise can defeat the entire control. Detection-only approaches see attacks but cannot prevent them.

Network Guard is designed so it does not become a target or landing point for an attacker. Every internal component is bounded by a permission model called 'object capabilities', where each component holds only the access rights it requires and can reach no resource beyond them. A component holding no capability to write to the OT-facing interface cannot do so. These boundaries are enforced by a mathematically proven kernel. An attacker who compromises a component cannot pivot to the OT network, because the boundaries preventing this are proven correct.

Network Guard is installed inline between an untrusted network and an OT device requiring protection. It parses every packet, applies configured inspection rules, logs what it observes, and acts on inspection verdicts with the ability to drop non-conformant traffic before it reaches the protected device. No modification to the protected device is required at any stage.

02 Problem statement

The challenge facing in-service OT equipment is not a single vulnerability but a structural condition, made up of three independent challenges: certification, exposure, and incident asymmetry. In combination, these leave certified devices exposed in ways conventional security controls cannot resolve.

The certification challenge

Certified OT equipment mostly cannot be modified without invalidating its certification. The scope of modification is broad: software changes, firmware updates, configuration alterations, addressing changes to the device's operational environment. The recertification process that follows typically takes months at significant cost.

Critically, certification locks in the vulnerabilities present at that point in time. A device certified five or ten years ago carries the security posture of its software stack as it existed then. Published CVEs against widely deployed platforms document exploitation paths that cannot be closed without modifying the device. CVE–2022–38465 (CVSS 9.3), for example, exposes the cryptographic keys protecting the Siemens SIMATIC S7–1200 and S7–1500 communication stack across millions of deployed controllers. At the time of writing, the device remains in service and the vulnerability remains open. The only available mitigations are network-layer controls external to the device.

Exposure trajectory

Connectivity to certified OT equipment continues to expand. Smart factory programs, predictive maintenance, remote diagnostics, and supply-chain integration all require OT systems to exchange data with IT networks and external platforms. These devices were not designed to authenticate command sources, validate input structure, or detect malformed traffic. An attacker coming from a compromised workstation can reach equipment directly with no internal defense and no ability to add one without revalidation.



Proof over probability

Asymmetry between detection and action

In IT environments, detection without prevention is often acceptable as compromises can be remediated post-hoc. In OT, that bound does not hold. A malicious command reaching a safety controller can cause physical damage, environmental release, or personnel harm before any responder can intervene.

Detection and logging can start a security architecture; inline protection completes it. Inline protection introduces its own risk: an appliance that can block traffic is a single point of failure. The same property that gives it protective value makes it an availability risk the design must address.

Summary of constraints

Any candidate solution must simultaneously:

- Require no changes to the protected device that would trigger product recertification.
- Provide inline protection capability when authorized, not simply detection alone.
- Address the availability risk of an inline appliance through architectural means.
- Be resistant to compromise itself against the threats it is deployed to mitigate.
- Operate in environments where qualified support is not on site.
- Be a retrofit addition to existing networks, not a replacement or migration.

These constraints are demanding individually and harder in combination.

03 Where Current Approaches Fall Short

The constraints identified in Section 2 define the requirements any solution must satisfy simultaneously. The approaches most often deployed against OT threats satisfy some of those constraints. None satisfies all of them.

Endpoint and device-level controls

“Just patch it” doesn’t work for OT. Modifying or patching firmware delivers direct protection but immediately invalidates certification. These controls are appropriate for general purpose IT endpoints, not for certified OT equipment.

Traffic logging

Network-level logging provides useful visibility but no ability to act on what is observed. In OT environments where a malicious command can cause physical consequence before remediation, an observation-only posture is a starting point and not the whole solution.

Delivering mirrored traffic also typically requires switch reconfiguration, which can itself be a certification touchpoint.

Inline firewalls and Intrusion Prevention System Appliances

Inline filtering provides the action capability that logging lacks, but introduces problems:

- **Availability**: the appliance becomes part of the data path; failure interrupts traffic with no architectural means to contain faults or recover automatically.
- **Security model**: on Linux or BSD, a single compromise can give control of the entire packet processing stack, enabling traffic manipulation against the protected device, potentially leaving you worse off than if no firewall or IPS had been installed.
- **Parser correctness**: inspection relies on conventional parsers whose correctness is tested rather than proven, leaving them susceptible to the crafted malformed packets that exploit OT protocol stacks.

Network segmentation alone

VLAN isolation and firewall zoning reduce exposure but do not address protocol-level threats reaching the protected device through legitimate network paths. Segmentation is necessary but not sufficient.

The gap Network Guard addresses

Network Guard addresses the gap these controls collectively leave: protection for certified OT equipment requiring no modification, with inspection grounded in formal proof rather than heuristic pattern-matching.





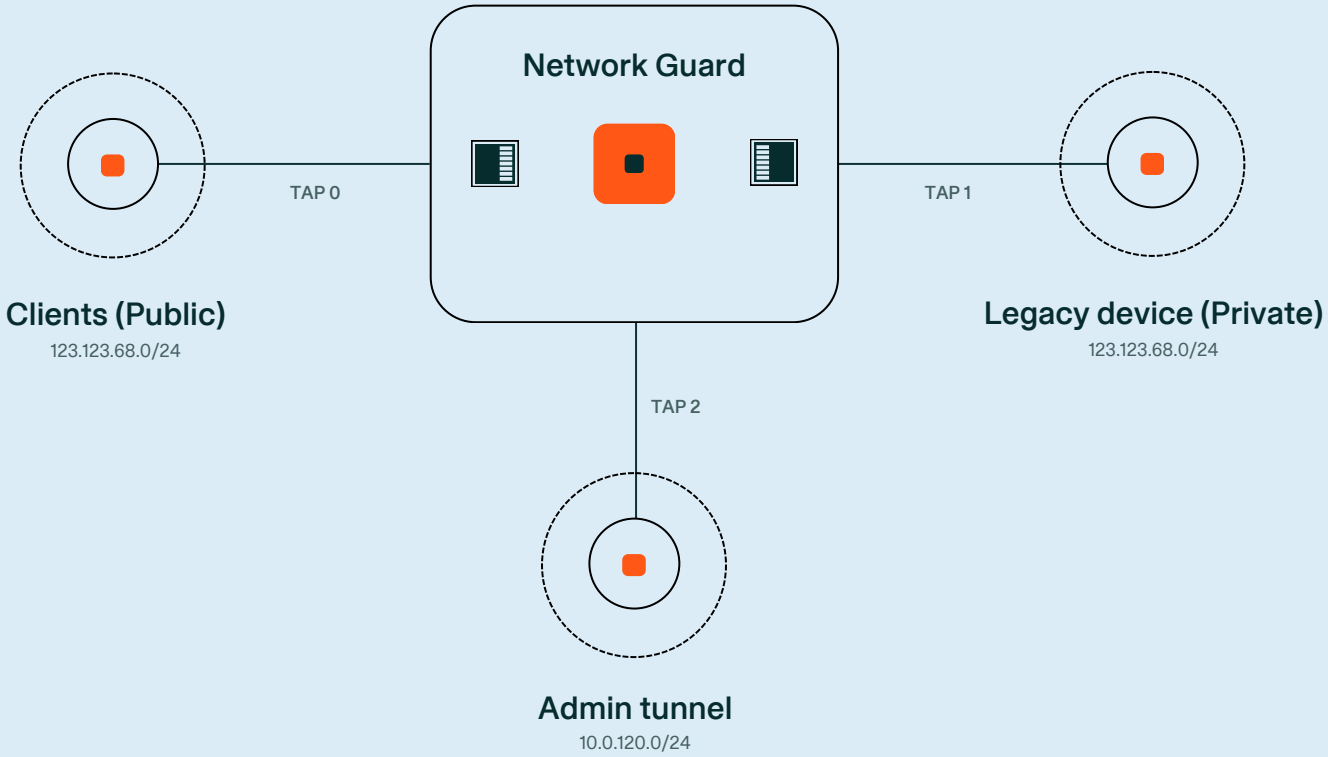
Building certainty into an uncertain world

04 The Network Guard Solution

The constraints in Section 2 leave one viable approach: a network-layer control, external to the certified device, that does not inherit the vulnerability model of the infrastructure it protects. Network Guard is that control.

Network Guard is deployed on the Compulab IOT-GATE-IMX8PLUS, a fanless industrial gateway with two 1Gbps RJ45 Ethernet ports and a power input. Rated for industrial temperature ranges (−40°C to +80°C) and with no moving parts. It can be mounted in a cabinet or on a DIN rail in the target environment.

One Ethernet port connects to the public-side network; the other connects to the protected device. All traffic in the single protected link passes between the two ports, through Network Guard. Network Guard is a retrofit addition: it is installed into an existing cable run and requires no network redesign or migration.

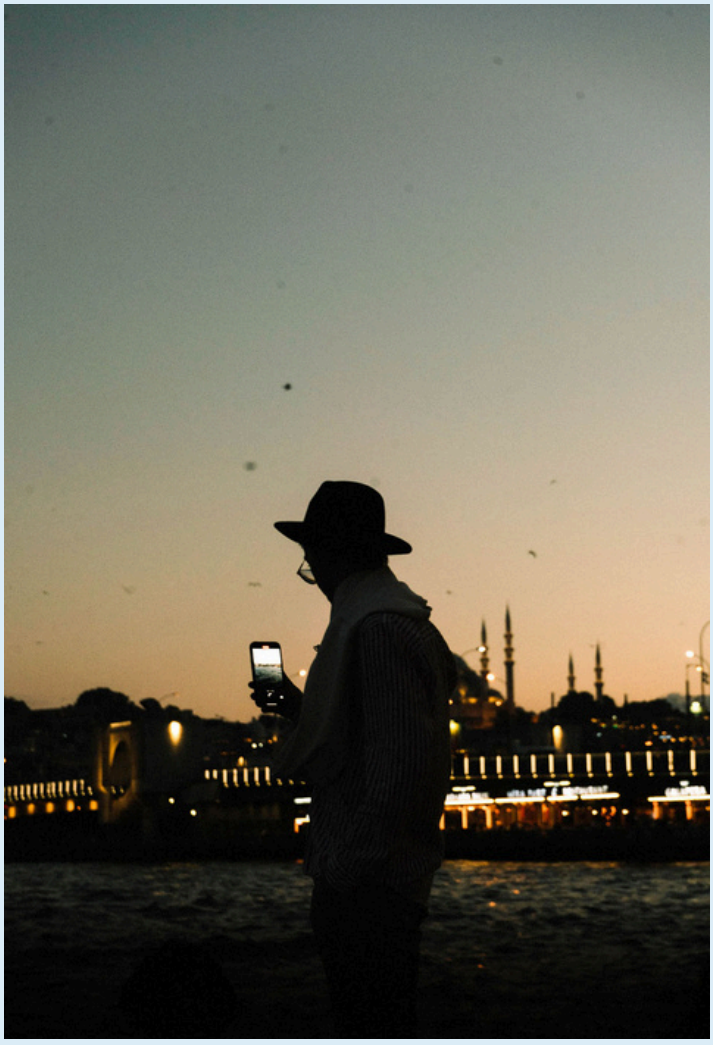


Packet processing architecture

Network Guard applies a consistent processing inspection pipeline to every packet that arrives on the public-side interface.

Ingress validation. Every packet is subject to deep packet inspection across three phases, each applying a different methodology and degree of introspection to address the gaps left by conventional validation mechanisms.

PHASE A	addresses a conventional firewall approach, establishing from the source and destination IPs and ports that the packet is permitted, but provides no assurance about protocol correctness or operational compliance.
PHASE B	uses formally verified parsers to establish that the packet's payload is correctly formed, a property that rule-based validation cannot provide.
PHASE C	establishes behavioral compliance with the operational baseline using policies.



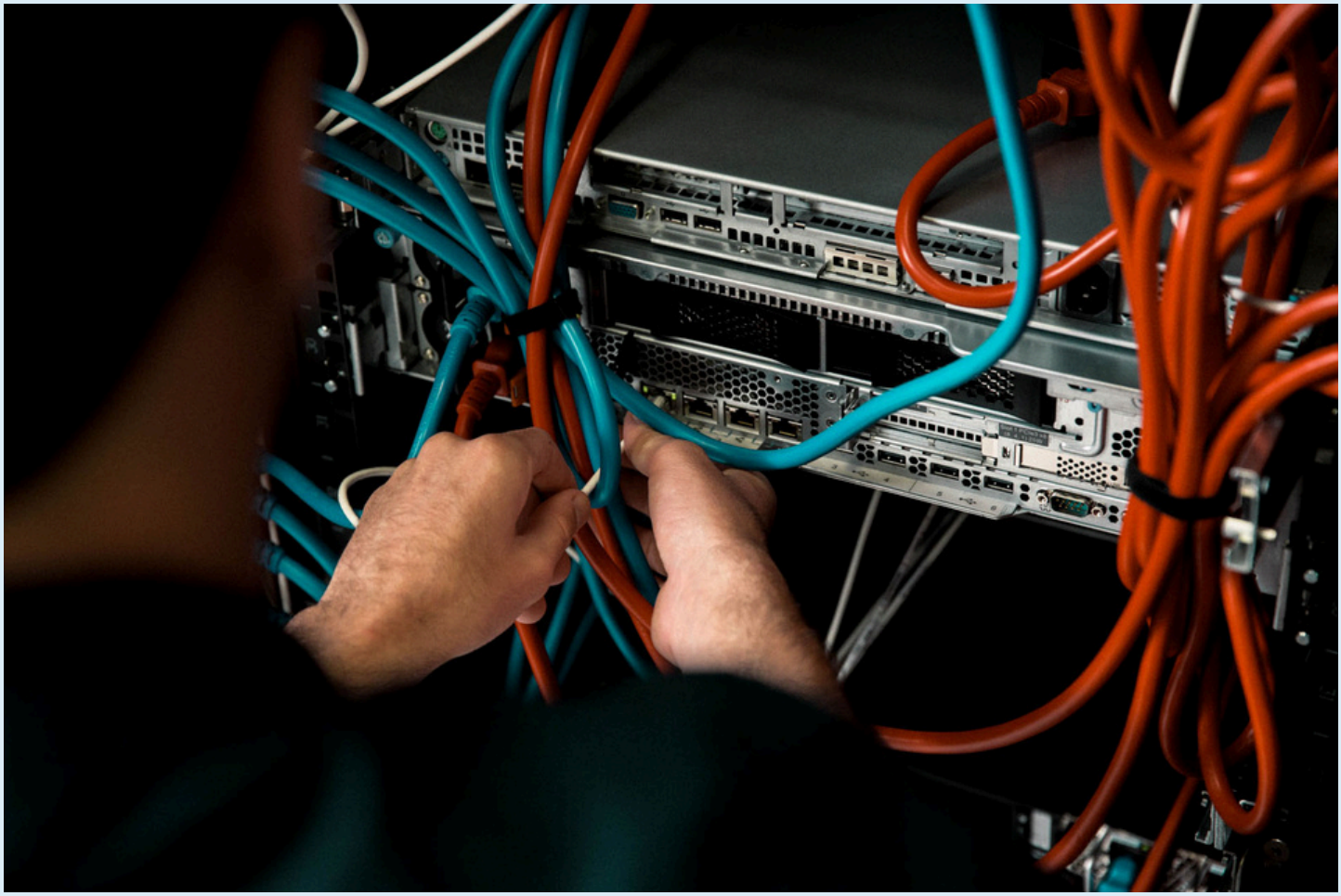
Filter analysis. Packets that pass ingress validation continue onward through the pipeline. Where no filter is assigned to the protocol, the packet is forwarded directly to the protected device unmodified. Where a filter is assigned, the packet is handed to that filter for further processing.

Decision and action. The filter returns a decision and action informed by the ingress validation: pass or drop the packet. All decisions are logged.

This pipeline is consistent regardless of which inspection phases or filters have been commissioned. Page 14 describes how inspection capability is built up incrementally across it.

What Network Guard does not do

Network Guard does not install software, drivers, or agents on the protected device. It does not modify the protected device's configuration, network addressing, or firmware. It does not change network topology beyond its own physical insertion.



Worked examples

Normal operation

A control system on the public network issues an instruction to a protected controller. The packet arrives at Network Guard's public-side port, passes ingress validation and is handed to the assigned filter. The filter finds the packet conformant and compliant. Network Guard forwards the packet to the protected controller and logs the transaction. From the perspective of the protected device and the control system, the network behaves as it did before Network Guard was installed.

Accidental malformation

A packet arrives with a corrupted transport-layer checksum. The cause is likely mundane rather than nefarious: electrical noise, a deteriorating connector, or a marginal switch port. The ingress validation stage recomputes the checksum and finds a mismatch. Packet is dropped from the pipeline before it reaches any further inspection or the protected device. A log entry records the source, timestamp, and nature of the violation.

Over time, a pattern of checksum errors from a specific interface can surface an underlying hardware fault that would otherwise be masked by retransmission behavior. The log is useful for infrastructure diagnostics and for threat modeling security.

Deliberate malformation

Crafted malformed packets are designed to trigger specific code paths in a target's firmware: buffer overflows or memory safety errors, type confusion, or state machine violations. These attack classes are behind a significant proportion of published CVEs against deployed OT equipment. The vulnerabilities exist because legacy devices' protocol stacks do not handle all possible inputs correctly.

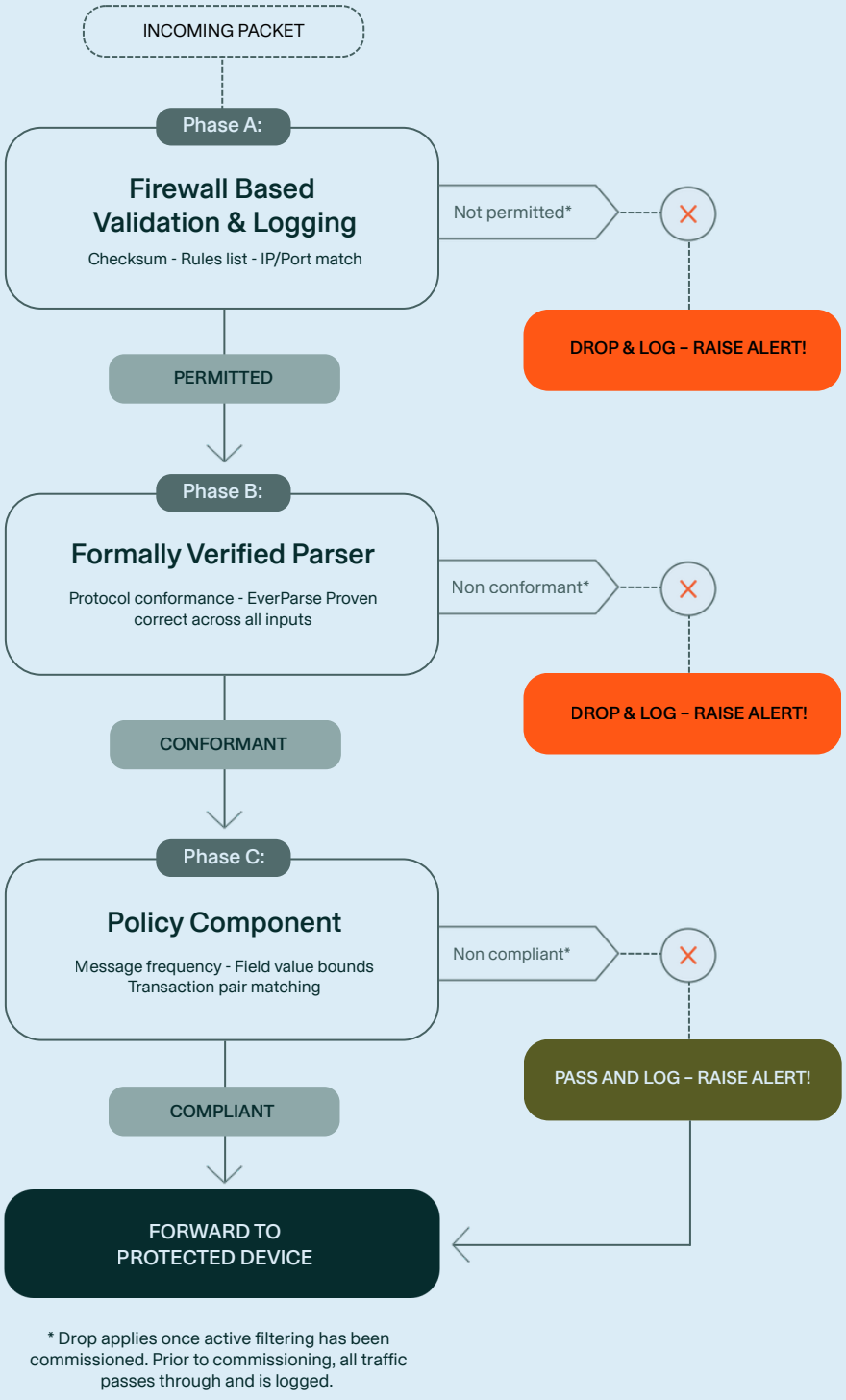
Network Guard's formally verified parsers address this. The parser's correct behavior across all possible inputs is proven against the official specification, not based on testing. A malformed packet crafted to exploit an edge case in a controller's firmware finds no such opportunity in a formally verified parser. If active filtering is enabled, the packet is dropped and logged. The protected device, which may carry the exact vulnerability targeted, never sees it.

Incremental adoption

Network Guard can be adopted incrementally, building inspection capability across the packet processing pipeline described on pages 11–12. Each phase can be commissioned and validated before the next is enabled, without disrupting traffic flow or requiring changes to the protected device, giving operational teams and accreditation authorities a structured path to active enforcement.

PHASE A	**Firewall based validation and logging** Source IP, destination IP, port, and protocol are checked against configured firewall rules. The frame checksum is verified. Each packet is assessed as permitted or not permitted. All traffic passes through and is logged. This phase establishes the traffic baseline and surfaces structural anomalies without protocol-specific knowledge.
PHASE B	**Protocol conformance with formally verified parsers** Beyond Phase A, each protocol message is validated for conformance to the relevant standard by a formally verified parser. Each packet is assessed as conformant or not conformant. All traffic continues to pass through and is logged. Each parser is commissioned for the protocol used in the target environment.
PHASE C	**Stateful inspection with policy compliance** Beyond Phase B, transactional pairs are matched for coherence, message frequency is checked against operator-defined tolerances, and field values can be inspected against configured operational bounds. Policy defines what is operationally compliant or not compliant, in addition to the protocol conformance of Phase B. All traffic continues to pass through and is logged. Policy bounds are operator-defined and can be adjusted remotely through Kry10's Fleet Server. A false positive is likely a configuration matter, not a fault.

Packet processing pipeline



* Drop applies once active filtering has been commissioned. Prior to commissioning, all traffic passes through and is logged.

Filter decision

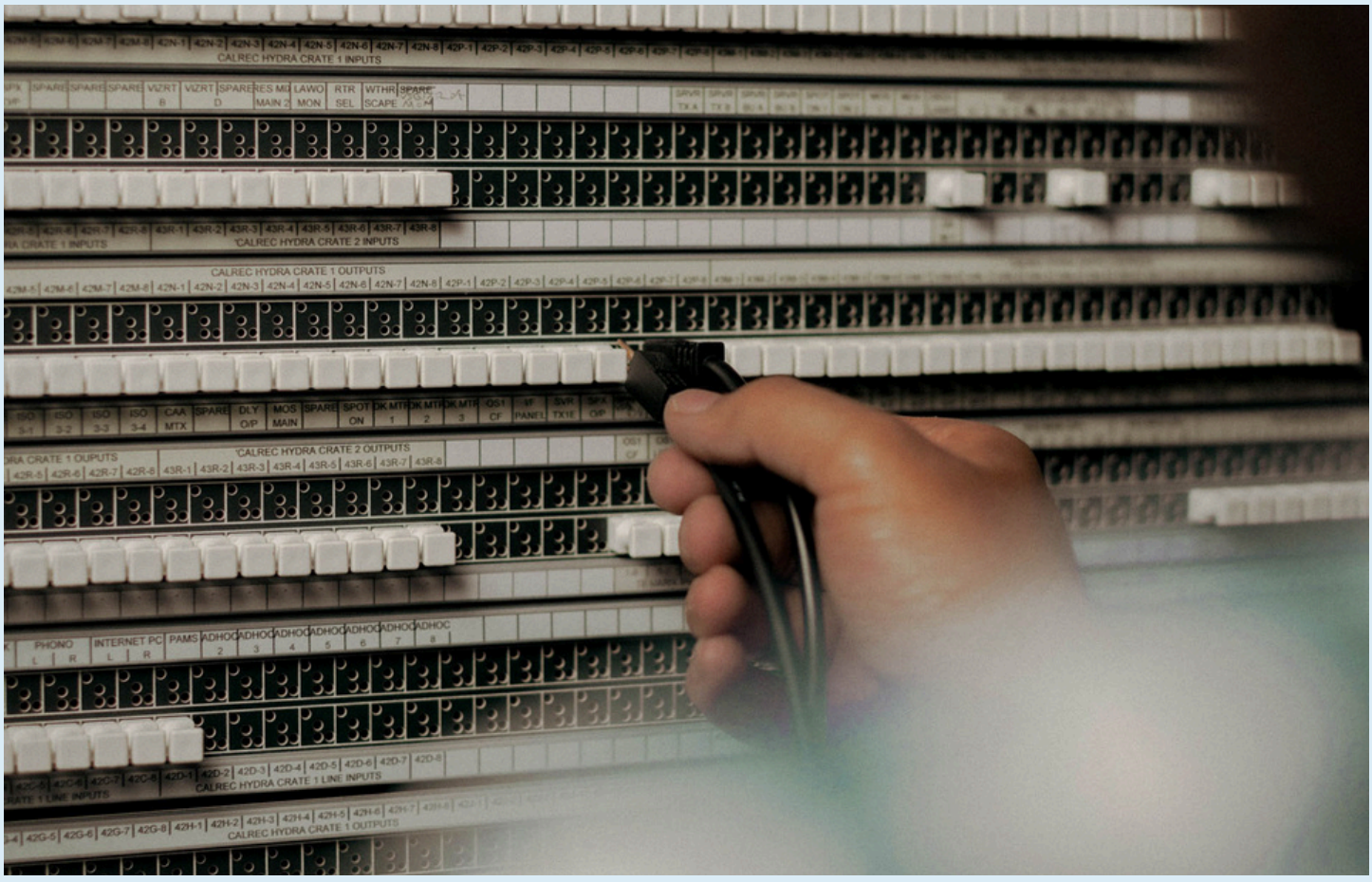
With Phase A, B, and C ingress validation established, the filter decision determines how traffic is handled. This example table below suggests how each Phase output can be applied once active filtering has been commissioned; prior to commissioning all traffic passes through regardless of inspection outcomes. All decisions are logged.

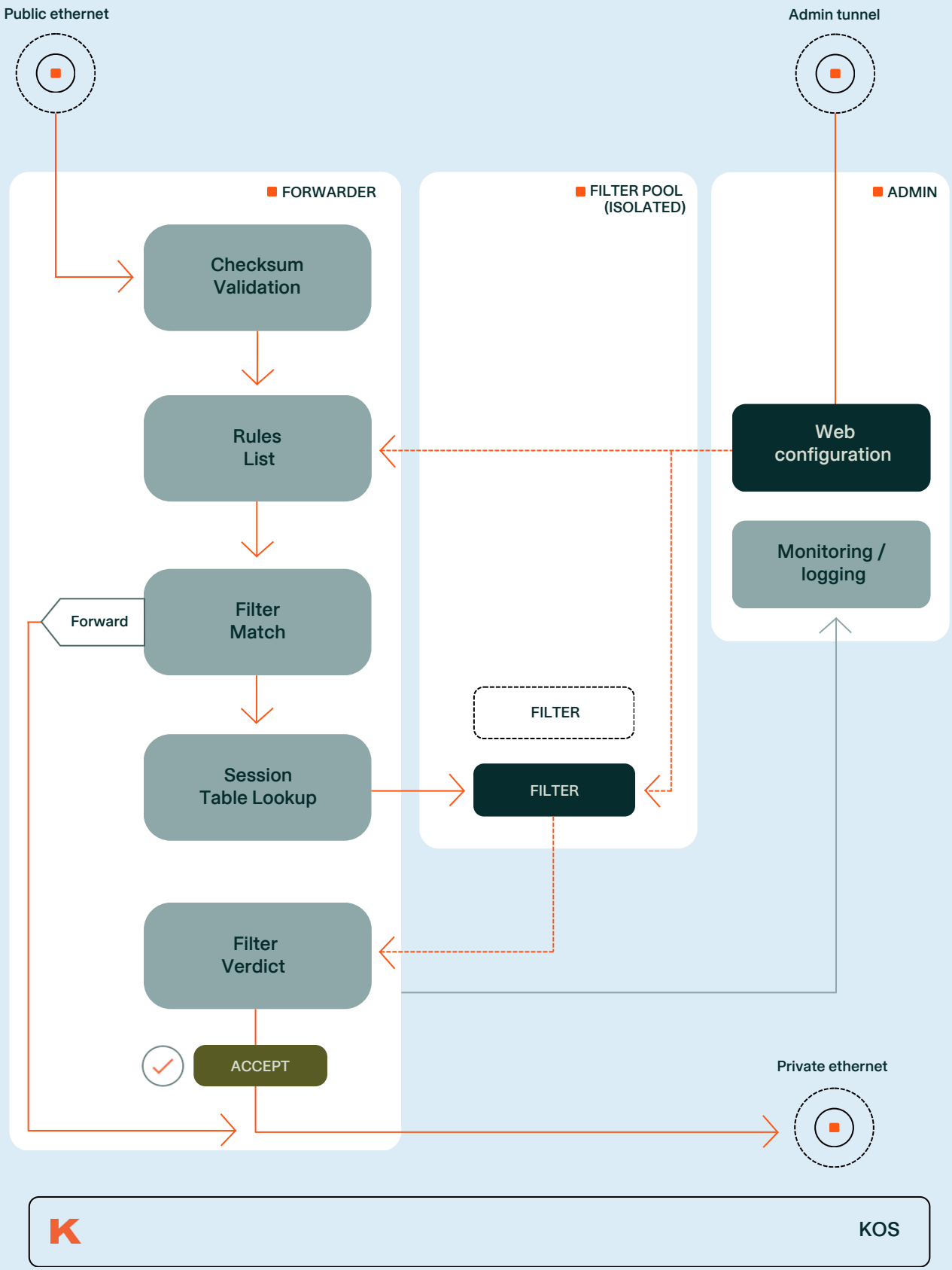
Transition to active filtering is applied on explicit client authorization and is fully reversible.

05 How Network Guard works

Network Guard Architecture

Network Guard delivers security properties through architecture. The formally verified kernel and formally verified parsers are mutually reinforcing: the kernel ensures correct isolation of Network Guard's internal components, and the parsers enforce tightly controlled communication, ensuring that a malformed packet cannot reach the device. Together they close the attack surface that conventional appliances leave open. The diagram below shows what that means in practice: three capability-bounded partitions on a formally verified foundation, where the boundaries between them are enforced by the kernel itself.




Public ethernet
Admin tunnel
FORWARDER
FILTER POOL (ISOLATED)
ADMIN
Checksum Validation
Rules List
Filter Match
Forward
Session Table Lookup
Filter Verdict
ACCEPT
FILTER
FILTER
Web configuration
Monitoring / logging
Private ethernet
KOS

The KOS substrate and isolation properties

Network Guard is built on KOS, founded on the seL4 microkernel. seL4 is the only operating system kernel whose isolation and security properties have been formally, mathematically proven. The kernel's behavior under all possible inputs is known.

The diagram on the previous page shows the three capability- bounded partitions KOS enforces at runtime. The Forwarder owns the packet processing path and holds capabilities to both network interfaces. The Filter Pool contains isolated inspection components, each bounded to a single assigned connection; inactive instances hold no live packet capabilities and a compromised active filter cannot affect any other. The Admin partition holds web configuration and monitoring/logging, connected via the encrypted Admin Tunnel.

- Components communicate only through a specialist type of permission called a capability, where possession gives access to the resource it governs: memory regions, hardware devices, communication endpoints, and kernel objects. No component can access a resource for which it does not hold a capability. There is no privilege escalation path.
- There is no concept of a "root" user in the KOS environment. Privilege is the set of capabilities a component holds. A component fully controlled by an attacker remains bounded by its capability set.

Practical consequences: on a Linux or BSD appliance, root compromise grants access to the entire packet processing stack and potentially permanent control of the device. On Network Guard, no equivalent boundary exists to exploit.

Formally Verified Parsers

The formally verified parser is one of Network Guard's core differentiators, and the layer where deployment-specific work is concentrated.

A conventional parser's correctness is established through testing. A formally verified parser's correctness is proven: its response to every structurally possible protocol input is demonstrated correct against the official standard. Network Guard's parsers are generated using EverParse, a formal verification framework developed by Microsoft Research that produces machine-checked proofs of memory safety and functional correctness from format specifications. There are no untested edge cases; the proof covers all inputs. Because protocol specifications are stable, the correctness guarantee is durable.

Threats evolve; the parser's conformance to the standard does not. New parsers can be commissioned for a specific protocol mix in the target deployment via Kry10's Fleet Server; physical access to Network Guard is not required to roll out these updates. We have already generated formally verified parsers for Modbus TCP, DNP3, and IEC 60870–5–104; integration into the filter is committed development. Sector-specific protocol parsers, including OPC–UA, IEC 61850, and CIP/EtherNet/IP, can be commissioned for development against the same interface.

The formally verified parser defines and enforces conformance to the protocol's standard. Policy compliance is a separately composed conventional component; it does not carry formal verification properties but is capability-isolated by KOS, bounding its blast radius in the same way as the parser. Both are deployed and configurable through Kry10's Fleet Server, independent of the protected device.

Management plane separation

Administrative access — configuration updates, log retrieval, policy changes, parser commissioning — is handled through Kry10's Fleet Server channel. Management traffic travels over the public-facing interface as a logically separated admin tunnel and does not traverse the forwarding path. The management interface holds no capability to write to the OT-facing network interface.

Fault behavior and availability

Other application-specific OT firewalls run on general-purpose operating systems and inherit the associated attack surface. A vulnerability in the underlying OS can undermine and disable any protection entirely. KOS's formally verified isolation prevents any fault from cascading across component boundaries.

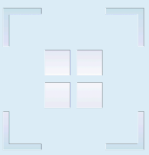
Network Guard is an inline appliance; a fault disabling the forwarding component will interrupt traffic until recovery. Two architectural properties mitigate this:

- **Structural containment.** A fault in any component is contained to that component by capability boundaries. Other components, the forwarding path, logging, and the management plane are unaffected by construction. Affected connections are automatically reset; all others continue uninterrupted.
- **Resilience by construction.** The forwarding path uses static resource allocation: no heap, no runtime memory allocation, no fragmentation. Memory exhaustion, allocator corruption, and heap vulnerabilities are absent by design.

The residual case divides by power state. Where power is maintained, Network Guard fails open: traffic continues to the protected device unprotected but uninterrupted while Network Guard restarts and reinitializes its capability-bounded partitions on KOS.

Loss of power interrupts traffic for the duration of the same boot and initialization sequence. No hardware bypass mechanism is currently available. The formally verified isolation of the forwarding path makes deliberate fault induction significantly harder than on a conventionally implemented appliance; the attack surface available to trigger a forwarder fault is correspondingly small.

What matters is the availability risk that Network Guard is deployed to prevent. Sophisticated attacks against critical OT infrastructure include state-sponsored prepositioning campaigns demonstrated by Volt Typhoon. The targeted exploitation of unpatched legacy devices causes operational disruption exceeding any interruption from a Network Guard restart. An unprotected device is itself an availability risk; the deployment question is not whether to accept risk, but which risk is greater.



06 Benefits and impact

The parsers and network interfaces in legacy OT devices weren't written with security in mind. Network Guard closes this gap: detecting and blocking packets that aren't conformant to the protocol standard (whether malicious or accidental), that aren't permitted by firewall rules, or that aren't compliant with operational policy.

For product security

Network Guard's security posture is statically auditable from the system manifest. Capability-enforced isolation and the absence of a privilege escalation path mean types of attack commonly found against (e.g.) Linux just don't exist. In the event of a vulnerability within a single component (e.g., due to third-party unverified software), any possibility of compromise is bounded by proven isolation.

For operations

The certified device's operational environment is unchanged; existing procedures remain valid. Network Guard is designed to operate without qualified support on site: parser updates are delivered remotely through Kry10's Fleet Server; no site visit is required to extend inspection to additional protocols. Component faults are handled automatically by the KOS supervision mechanism. Recovery from severe faults requires a power cycle at worst.

For risk and compliance

Network Guard's logging components have no ability to write traffic toward the protected device, nor can they block packet forwarding. Log integrity is protected at two levels:

- **Within the appliance** — the logging component occupies its own KOS partition and a compromised filter or parser cannot tamper with the forwarder's independent log record;
- **In transit** — the logging component holds its own dedicated encrypted endpoint on Kry10's Fleet Server tunnel, transmitting log data under a cryptographically independent channel from management traffic.

Transition to active enforcement requires explicit client authorization following review of baseline data. The principal residual risk is Network Guard's inline position. The description of mitigations can be found on page 19.

For procurement

Introduction of Network Guard as a new network element does affect the system accreditation context. However, the certified device being protected is unchanged; recertification is not triggered.

The architectural facts presented here as inputs required for accreditation engagement are: **no modification** to the protected and certified device, Network Guard operation is as a **transparent upstream addition**.

Where an authority considers the network layer part of the device 's operational environment, Network Guard is designed to support accreditation engagement at this level.



07 Broad sector application

Network Guard’s architecture addresses a constraint that appears across critical infrastructure sectors. The formally verified parser commissioning model serves different sectors through different parser deployments on the same appliance and fleet infrastructure. Network Guard can be readily configured and commissioned to protect equipment across a wide variety of sectors with specific communication and compliance requirements.

08 Network Guard sector application

Sector	Equipment under protection	Typical commissioned filters	Compliance framework
MANUFACTURING & PROCESS	Safety PLCs, process controllers	Modbus TCP, OPC–UA, CIP	IEC 62443, IEC 61508
ENERGY & UTILITIES	RTUs, SCADA front-ends, protective relays	DNP3, IEC 61850, IEC 60870–5–104	NERC CIP, IEC 62443
TRANSPORT & RAIL	Signaling controllers, interlocking systems	Vendor-specific signaling protocols	CENELEC EN 50128 / 50129
HEALTHCARE	Medical devices, lab instruments	HL7, DICOM, vendor-specific	FDA 510(k), MDR
AEROSPACE GROUND	Test rigs, certified ground support equipment	Vendor-specific ARINC-family	DO–254, DO–178C
DEFENSE	Mission systems, in-service platforms	Mission-specific protocols	Various national accreditation

In each case the protected asset is not modified and faults are contained at component level by capability boundaries. The availability trade-off is known and bounded, which is a stronger position than unmitigated exposure to unpatched vulnerabilities in certified devices that cannot be remediated without recertification. Each sector should assess this against the availability risk of leaving certified devices unprotected.



Life expects things *to work*

09 Conclusion



Why conventional approaches do not fit OT

Conventional firewalls and intrusion prevention systems (IPS) inherit IT assumptions: general-purpose operating systems, privilege models that collapse at root, and fault models mismatched to OT operational realities. Other existing application-specific OT firewalls introduce some protocol-specific inspection but retain OS-level vulnerability exposure and reliance on testing, rather than proven parser correctness.

What Network Guard provides

Network Guard is installed inline, records what it sees, inspects packets against formally verified parsers, and forwards permitted traffic. It requires no modification of the protected device. Components are isolated by capabilities enforced by a proven kernel. KOS's supervision mechanism provides automatic recovery without operator intervention. Protocol-specific inspection is commissioned per sector and extended remotely as requirements evolve.

The availability trade-off

Network Guard's inline position introduces an availability consideration shared by every inline security appliance. Static allocation, capability isolation, and automatic recovery make appliance-level faults rare. The residual case requires tolerance of occasional brief interruption; we believe this risk is substantially lower than remaining unprotected.

Considerations for procurement

Network Guard is in active development. The core appliance, KOS and Kry10 Fleet Server management are available today. Committed development and funding to deliver a suite of integrated formally verified parsers with filters for manufacturing and energy sectors has been secured. Real-world impact introduced by inline inspection relative to the target environment will need to be addressed during engagement.

Next steps

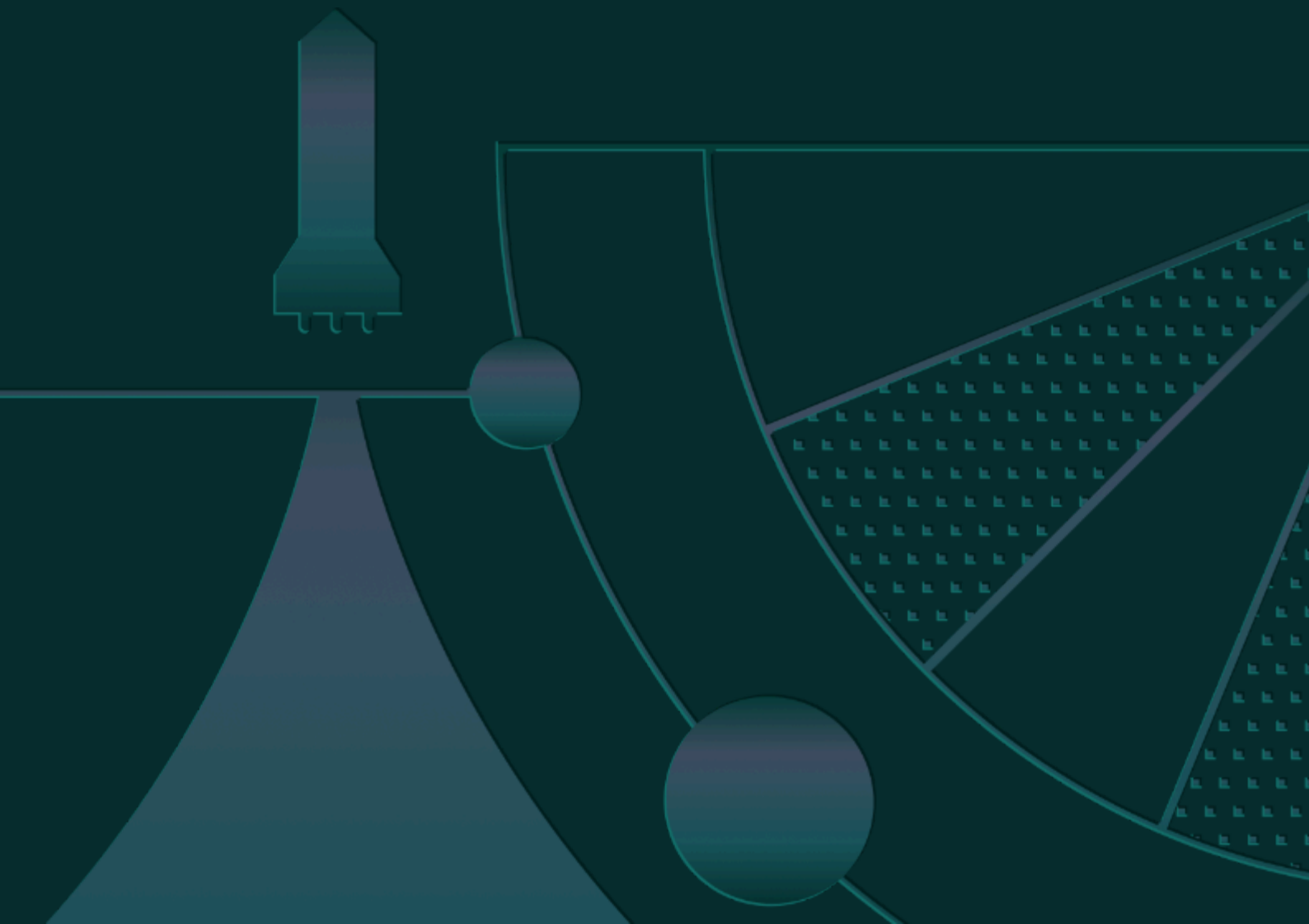
Certified devices in field operation carry vulnerabilities that cannot be closed without triggering revalidation, and connectivity demands continue to expand their exposure. Network Guard provides structured protection that is transparent to the certified device, grounded in formal assurance, and incrementally adoptable at a pace determined by the operator and the accreditation authority.

The appropriate starting point is a scoped initial deployment on a defined network segment. The first phase requires no commitment to active filtering and produces immediate value: an asset inventory, a traffic baseline, and an evidence record that informs every subsequent decision. This data supports accreditation processes, and brings insight to broader reduction of vulnerabilities across OT networks.

If your organization operates OT networks and hasn't yet established a threat visibility baseline, this is the lowest-risk starting point available.

We'd welcome the conversation. Reach us at:

KRY10.COM
HELLO@KRY10.COM
USA/UK/AUS/NZ



REFERENCES

1. seL4® Foundation. The seL4® Microkernel — formal verification record. sel4.systems
2. NXP Semiconductors. i.MX 8M Plus Applications Processor — product documentation.
3. Compulab. IOT-GATE-IMX8PLUS — industrial gateway product specification.
4. IEC 62443 series — Industrial communication networks — Network and system security.
5. IEC 61508 — Functional safety of electrical/electronic/programmable electronic safety related systems.
6. Siemens ProductCERT. CVE-2022-38465 — SIMATIC S7-1200 and S7-1500 symmetric protection key vulnerability. CVSS v3 Base Score: 9.3. nvd.nist.gov/vuln/detail/CVE- 2022-38465
7. Microsoft Research. EverParse: Verified secure zero-copy parsers for authenticated message formats. project-everest.github.io/everparse/
8. Cloudflare. BoringTun: A userspace WireGuard implementation. github.com/cloudflare/boringtun
9. US Cybersecurity and Infrastructure Security Agency (CISA), NSA, FBI et al. Advisory AA24-038A: VoltTyphoon PRC State-Sponsored Actors Compromise and Maintain Persistent Access to U.S. Critical Infrastructure. February 2024. www.cisa.gov/newsevents/cybersecurity-advisories/ aa24-038a

We’re ready to *partner*

Let us help you strengthen what matters

KRY10.COM
HELLO@KRY10.COM
USA/UK/AUS/NZ

KRY10