Wire-Level Intelligence: How WireTrace Analyzes Network Traffic

A technical deep-dive into WireTrace's protocol intelligence architecture, multi-signal classification engine, behavioral baseline methodology, and threat detection approach. This paper explains how passive traffic analysis produces asset inventory, security insights, compliance evidence, and threat detection across IT, OT, IoMT, and IoT environments.

WireTrace Technical Whitepaper | Version 1.3 | 2026

1. Introduction

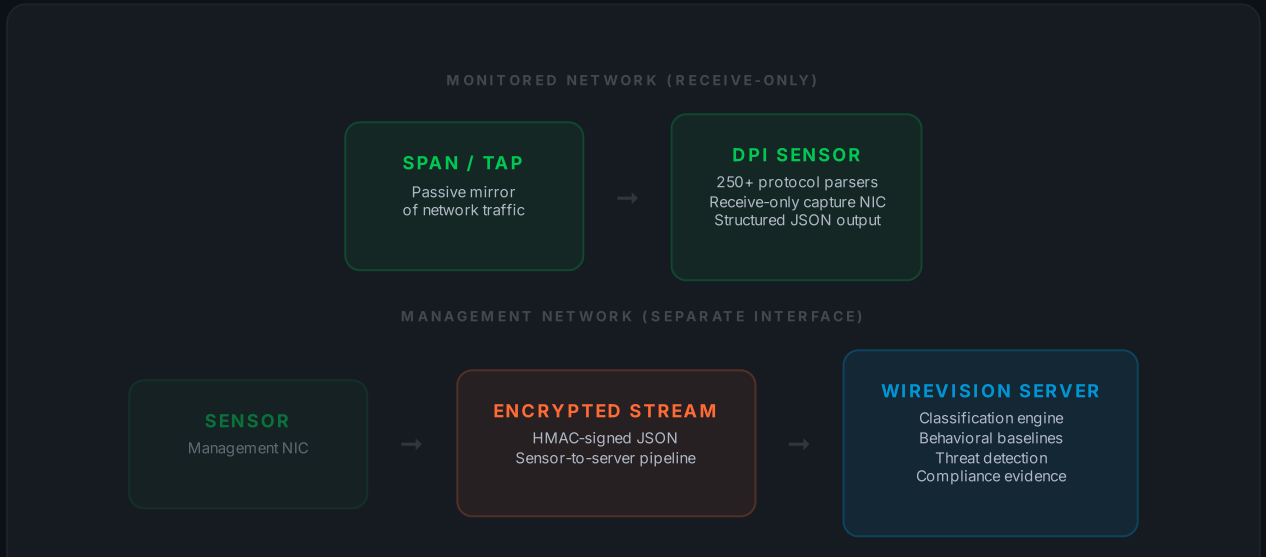
Network visibility has traditionally relied on two approaches: agent-based telemetry (requiring software on every endpoint) and active scanning (sending probes to discover devices and vulnerabilities). Both approaches fail in environments where agents cannot be installed (industrial controllers, medical devices, IoT) or where active probing creates operational risk (safety-critical OT networks, clinical environments).

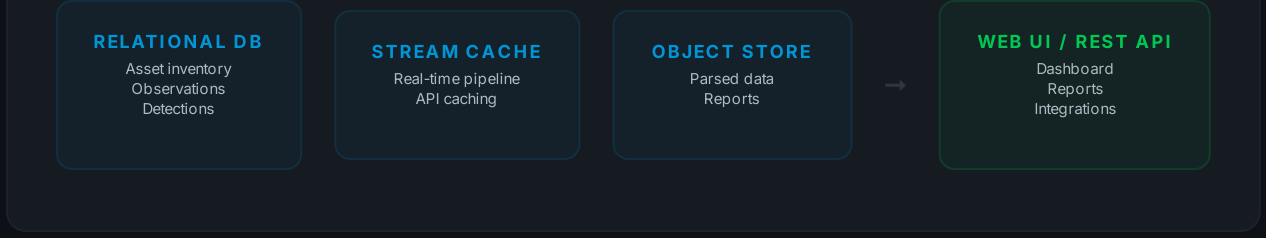
WireTrace takes a fundamentally different approach: it builds on passive deep packet inspection at the wire level and extends it into Deep Protocol & Asset Intelligence (DPAI) - protocol semantics plus asset identity. By analyzing a copy of network traffic from a SPAN port or network TAP, WireTrace extracts device identity, protocol commands, certificates, behavioral patterns, and security findings. The sensor operates in receive-only mode on its capture interface - zero packets are transmitted onto the monitored network. Sensor-to-server communication occurs over a separate management interface, and optional SNMP enrichment and Bahith active scanning, when an operator enables them, transmit from the management interface only and are off by default.

This paper describes the technical architecture and methodology behind WireTrace's core intelligence engines: Protocol Intelligence, Asset Classification, Behavioral Baselines, Threat Detection, and Compliance Evidence Generation. It also addresses visibility boundaries, encrypted traffic handling, and deployment considerations.

2. Platform Architecture

WireTrace uses a distributed sensor-server architecture. Sensors capture and parse traffic at the network edge; a centralized server performs classification, analytics, threat detection, and compliance evidence generation.





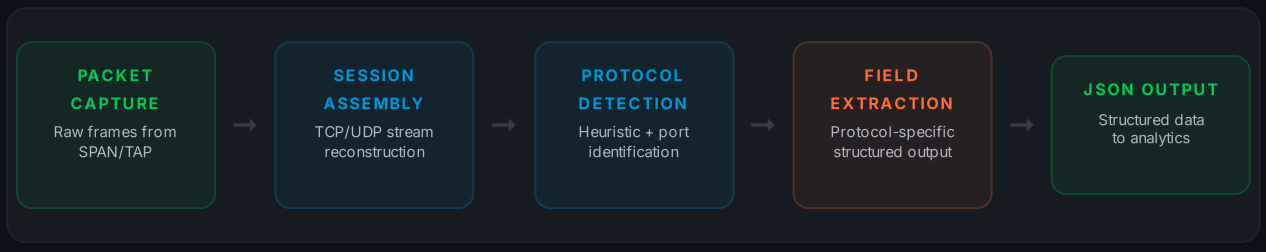
Capture isolation: The sensor uses two separate network interfaces. The capture NIC connects to the SPAN/TAP port in receive-only mode and never transmits. The management NIC connects to the WireVision server on a separate network path. This ensures zero impact on the monitored network.

3. Protocol Intelligence Engine

The foundation of WireTrace is its deep packet inspection engine - a high-performance native parser library that decodes network protocols at the application layer. Unlike port-based identification (which maps port numbers to assumed services) or signature matching (which looks for known byte patterns), WireTrace performs application-layer protocol dissection for supported protocols - parsing each protocol's grammar to extract structured fields from unencrypted protocol sessions.

3.1 Parsing Architecture

The DPI engine operates in a pipeline: raw packets are captured from the SPAN/TAP interface, reassembled into protocol sessions, and passed through protocol-specific parsers. Each parser understands the protocol's message structure and extracts fields relevant to asset identification, security analysis, and operational monitoring.



3.2 Protocol Depth Matrix

WireTrace supports 250+ protocols across four domains. The depth of extraction varies by protocol type and encryption state:

PROTOCOL	EXTRACTED FIELDS	VISIBILITY
Modbus TCP/RTU	Unit ID, function codes, register addresses and values, read/write discrimination, request/response correlation	Full
S7Comm	PLC model and firmware (via SZL), programming activity, diagnostic access, vendor extraction	Full
DNP3	Master/outstation roles, control relay commands, analog/binary data, unsolicited responses, outstation addressing	Full
EtherNet/IP CIP	Device identity (ListIdentity), CIP service codes, connection paths, I/O data	Full
DICOM / HL7	Imaging commands, clinical messaging, device associations, study metadata	Full
Medical (Philips, Draeger, GE, etc.)	Device manufacturer and model, clinical function, telemetry patterns	Full
TLS 1.2 and earlier	Certificate subject, issuer, validity, key strength, chain integrity, cipher suite, SNI	Full
TLS 1.3	SNI (ClientHello), negotiated cipher suite, JA3/JA4 fingerprints, session flow metadata	Partial - certificate encrypted
SSH	Protocol version, key exchange algorithms, host key fingerprint, session flow	Partial - payload encrypted
DNS / DHCP / LLDP / SSDP	Hostnames, queries, responses, device announcements, switch port mapping	Full
HTTP	User-Agent, Host, URI, methods, response codes, content types	Full
IPsec ESP / VPN tunnels	Endpoints, flow metadata, SPI values	Metadata only

Key distinction: Port-based identification tells you "something is communicating on port 502." WireTrace tells you "Unit ID 1 sent a Write Multiple Registers command to addresses 40001-40010 with values [0x00FF, 0x0100, ...]." This level of detail is what enables meaningful security analysis in industrial and clinical environments.

4. Encrypted Traffic and Visibility Boundaries

Passive deep packet inspection operates on a copy of network traffic as it appears on the wire. This means the depth of analysis depends on whether the application payload is encrypted.

Full Visibility (Cleartext Protocols)

For unencrypted protocols - including most industrial protocols (Modbus, S7Comm, DNP3, EtherNet/IP, BACnet, OPC-UA in Security Mode None), medical device protocols (DICOM, HL7, proprietary vendor protocols), and enterprise protocols (HTTP, FTP, SMTP, Telnet, DNS, DHCP, SNMP, LLDP, SSDP) - WireTrace extracts full application-layer fields, commands, parameters, and identity data.

Partial Visibility (Encrypted Sessions)

For encrypted sessions, WireTrace extracts observable handshake metadata and flow indicators. For TLS 1.2 and earlier, the server certificate is transmitted in cleartext during the handshake, providing full certificate field extraction. For TLS 1.3, the Certificate message is encrypted - WireTrace captures SNI (from ClientHello, unless Encrypted Client Hello is used), negotiated cipher suite, JA3/JA4 fingerprints, and session flow metadata. SSH, IPsec ESP, and VPN tunnel payloads are opaque to passive observation; only endpoints, flow timing, and protocol metadata are available.

Visibility note: Passive monitoring discovers devices based on observed network communication. Devices that are powered off, communicate exclusively through encrypted tunnels with no observable cleartext sessions, or are on network segments not covered by a SPAN/TAP may not be fully discovered or classified. SPAN port configuration, oversubscription, asymmetric routing, and VLAN coverage affect the completeness of traffic visibility. WireTrace complements - but does not replace - other asset inventory sources for comprehensive coverage.

5. Multi-Signal Asset Classification

Asset classification in WireTrace is performed by a weighted voting engine that combines multiple independent signals to produce a confident device identification. This approach is fundamentally different from single-signal methods (MAC OUI lookup, banner grabbing, or agent-based fingerprinting).

5.1 Classification Signals

Each observed device accumulates evidence from multiple sources:

- **Protocol behavior:** Which protocols the device speaks, how it speaks them, and what role it plays (client/server, master/outstation, publisher/subscriber)
- **DPI-extracted identity:** Vendor and model information parsed from protocol fields (S7Comm SZL data, ENIP ListIdentity, PROFINET DCP, LLDP system description)
- **MAC OUI:** Vendor prefix from the device's MAC address - used as supporting evidence, not as the sole classifier
- **DHCP fingerprint:** Hostname, vendor class, requested options from DHCP transactions
- **mDNS/SSDP/NBNS:** Service announcements, device descriptions, model identifiers from discovery protocols
- **TLS handshake data:** Certificate fields from TLS 1.2 connections; SNI, cipher suite, and JA3 fingerprint from all TLS versions
- **HTTP User-Agent:** Browser, OS, and application identification from HTTP headers
- **Behavioral patterns:** Communication timing, protocol mix, peer relationships, and data volume patterns

5.2 Weighted Voting

Each classification signal carries a priority weight (1-200) based on its reliability and specificity. DPI-extracted identity from industrial protocols (e.g., S7Comm SZL returning "Siemens S7-1200") carries a high priority (100+), while MAC OUI (which only identifies the vendor, not the model or function) carries a lower priority (50).

Multiple rules can vote on the same device. The classification engine resolves conflicts by selecting the highest-priority match, with tie-breaking based on signal specificity. This means a Philips ventilator behind a VMware virtual NIC is correctly

classified as a medical device - not as a VMware server - because the proprietary Philips protocol fingerprint outweighs the MAC OUI.

5.3 Classification Rule Structure

Classification rules are defined in YAML and organized by domain (industrial, medical, enterprise, IoT). Each rule specifies match conditions (which protocols, which field values, which OUI prefixes) and outputs (device type, vendor, model, Purdue level, domain). As of version 1.3, 427 classification rules across 13 domain-specific rule files cover industrial controllers, medical devices, enterprise infrastructure, and IoT endpoints.

6. Behavioral Baseline Engine

WireTrace establishes per-device, per-protocol behavioral baselines from observed traffic. Baselines capture what is "normal" for each device - which protocols it uses, which peers it communicates with, how often, with what command patterns, and at what data volumes.

6.1 Baseline Components

Communication Baselines

For each device: known communication peers, protocols used per peer, first/last seen timestamps, typical data volumes, and communication frequency. New peers, new protocols, or significant volume changes trigger deviation alerts.

Command Baselines

For industrial protocols: which function codes are normally used, which register ranges are read/written, polling intervals, and request/response patterns. Unusual write commands, new register accesses, or frequency changes are flagged.

6.2 Learning Period and Refinement

Initial behavioral baselines begin forming within 24-48 hours of continuous observation, which is sufficient for steady-state OT environments with consistent polling patterns. The baseline engine refines continuously as more operational history is accumulated, incorporating weekly cycles, shift patterns, and maintenance windows over the first 7-14 days. Environments with shift-based manufacturing, seasonal HVAC operations, or batch processing benefit from extended observation periods for full cycle coverage.

Deviations are reported with full protocol context - the alert shows what changed, what the baseline value was, and the specific protocol evidence. This context enables analysts to rapidly distinguish genuine anomalies from expected operational variations.

EXAMPLE: ANOMALOUS MODBUS WRITE DETECTED

1. **Observed:** Modbus TCP packet from `10.1.2.50` to PLC at `10.1.1.10` - Function Code 16 (Write Multiple Registers), addresses `40001-40010`
2. **Parsed fields:** Unit ID: 1, FC: 16, Start Address: 40001, Quantity: 10, Values: [0x00FF, 0x0100, ...]
3. **Baseline check:** Device `10.1.1.10` (Siemens S7-1200, classified via S7Comm SZL) has a baseline of FC 3 (Read Holding Registers) from `10.1.2.50` every 2 seconds. FC 16 (Write) has never been observed from this source.
4. **Alert generated:** "Anomalous Modbus Write - PLC 10.1.1.10 received Write Multiple Registers (FC 16) from engineering workstation 10.1.2.50. Baseline: Read-only (FC 3). This source has never written to this PLC."

7. Security Insights Engine

The Security Insights engine continuously analyzes observed traffic for security findings across several categories:

Cryptographic Posture

TLS handshake analysis extracts certificate fields (subject, issuer, validity, key strength) from TLS 1.2 and earlier connections. For TLS 1.3, WireTrace captures SNI, negotiated cipher suite, and JA3/JA4 fingerprints. Self-signed and expired certificate detection. Weak cipher suite

Exposure Findings

Cleartext credentials in HTTP, FTP, SMTP, Telnet. Exposed management interfaces (SSH, RDP, SNMP with weak community strings). Unprotected industrial protocols (Modbus without authentication, S7Comm in programming mode). Unencrypted database connections. These are

negotiation (observable from handshake regardless of TLS version). Cleartext protocol usage where encryption should be expected.

observed exposures - high-confidence findings based on what is actually seen on the wire.

8. Threat Detection

WireTrace's threat detection combines three approaches: indicator matching, behavioral anomaly detection, and attack surface analysis.

- **IoC Matching:** Observed IPs, domains, and URLs matched against threat intelligence feeds (STIX/TAXII). Infrastructure whitelist prevents false positives from known-good services (e.g., CDN providers, cloud infrastructure, OS update servers). Per-IoC exclusion allows analysts to dismiss confirmed-benign indicators.
- **Behavioral Detection:** Deviations from established baselines - new communication peers, unusual protocols, command pattern changes, volume anomalies - surfaced with full protocol context and baseline comparison data.
- **Attack Surface Scoring:** Per-asset risk score based on observed exposures, categorized by severity: cleartext credential transmission (critical), expired or self-signed TLS certificates (high), exposed management interfaces with weak authentication (high), and unprotected industrial protocol access (medium-high). Scores reflect what is actually observed on the wire. CVE correlations based on observed firmware versions are probabilistic and should be validated through targeted assessment.

9. Compliance Evidence

WireTrace generates supporting evidence for selected controls within 7 frameworks - IEC 62443, ISO 27001, HIPAA, NCA ECC, NCA OTCC, NIST CSF 2.0, and NERC CIP - plus custom frameworks. This evidence is derived from observed network behavior - asset inventories from protocol communications, segmentation validation from documented cross-zone traffic, encryption posture from TLS handshake analysis, and access control patterns from communication flow records.

Traffic-derived evidence supports audit preparation and reduces manual evidence collection. It does not constitute compliance certification and does not replace all evidence requirements for any framework. Controls that require policy documentation, physical security measures, or organizational processes are outside the scope of passive network observation.

10. Limitations and Scope

Encrypted Payloads

Application-layer content within TLS 1.3, SSH, IPsec ESP, and VPN tunnels is not visible to passive observation. Handshake metadata, flow behavior, and endpoint identifiers remain available.

SPAN/TAP Coverage

Visibility depends on correct SPAN port configuration, adequate mirror capacity (oversubscription can drop packets), symmetric routing coverage, and VLAN mirror scope. VM-to-VM traffic on the same hypervisor may not reach a physical SPAN port.

Device Discovery Scope

Passive monitoring discovers devices that generate observable network traffic on monitored segments. Devices that are powered off, air-gapped from monitored segments, or communicate exclusively through opaque tunnels may not be discovered.

11. Deployment

All components run as containers. Typical server installation completes within 10-15 minutes using the self-extracting installer on recommended hardware. Sensors enroll via activation token and begin transmitting parsed protocol data within 60 seconds. The platform operates fully on-premises with no cloud dependency. Air-gapped deployment is supported; vulnerability feed updates can be performed via offline file transfer.

Request a Technical Evaluation

A WireTrace proof-of-value deployment includes one or more sensors on your network, 2-4 weeks of continuous observation, a full asset inventory with classification, security findings report, and a compliance evidence summary.

wiretrace.io | sales@wiretrace.io