Security by Default, Legacy by Deployment: The Encryption Gap in the Installed S7 Base

Bruno Salmazo

Liscere

July 2026 · Liscere Briefing · LB-2026-04 · v1

Executive Summary

Siemens S7 communication has moved decisively towards encryption by default. Current controllers, configured with the current engineering toolchain, protect programming and HMI traffic with certificate-based transport security as a native, preconfigured feature. This briefing documents a gap between that norm and the field: the installed base of S7 controllers remains, in its majority, communication that a passive observer can read. Three regimes coexist across the estate, from cleartext S7comm on the widely deployed S7-300 and S7-400 families, through application-layer encrypted S7CommPlus, to S7CommPlus wrapped in TLS on current hardware. The field is dominated by the readable regimes, and the gap persists through economics rather than ignorance. For any approach that passively observes S7 traffic, the implication is mixed rather than bleak: across most of the deployed estate, passive observation remains viable because the traffic is readable; where encryption is in force, it marks the boundary of the passive vantage. Encryption is the state of the art in S7 communication. It is not yet the state of the field.

1. The three encryption regimes of S7

Siemens S7 communication is not a single protocol with a single security posture. Across the installed base, three distinct regimes coexist, and conflating them is the most common source of error when reasoning about what an observer can and cannot recover.

Cleartext (S7comm, protocol ID 0x32). The original S7 communication protocol, in service since the mid-1990s, carries its payload in the clear [1]. It provides no transport encryption and no authentication of the peer; where a controller password is configured, it is exchanged in cleartext within the protocol's userdata function [1]. This is the protocol of the S7-300 and S7-400 families, and of the earliest S7-1200 firmware. Its structure is well understood and openly dissected: a message-type field, known as ROSCTR, distinguishes job requests from responses, and a function code within each request identifies read, write, and control operations against named memory areas [1]. Any party with a vantage point on the traffic recovers the operations and their operands directly.

Legacy-protected (S7CommPlus, protocol ID 0x72, pre-TLS). Introduced with the S7-1500 family and later S7-1200 firmware, S7CommPlus was Siemens' response to the authentication gap that Stuxnet exposed [2]. It encrypts the application payload under a proprietary scheme, attaches a per-message integrity value, and derives session keys that render recorded traffic unusable for replay [1]. The confidentiality of the older configuration data and of legacy programming and HMI communication rested, however, on a private key built into the CPU family rather than provisioned per device. That design was later shown to be insufficient: the global key could be recovered by an offline attack against a single controller of the family, a vulnerability assigned CVE-2022-38465 with a CVSS base score of 9.3, reported by Claroty's research team and coordinated through CISA [3][4].

TLS-wrapped (S7CommPlus over TLS). From TIA Portal V17 onwards, on controllers meeting the required firmware level, S7CommPlus is carried inside a TLS session with certificate-based key management [5]. This replaces the shared-key legacy protection with per-device certificates and standard transport security. It is the regime configured on current-generation hardware with the current toolchain.

The distinction between the second and third regimes matters and is easily lost. The S7CommPlus application layer carries its own encryption and integrity mechanisms independently of TLS; the TLS wrapper is a separate, transport-level layer added later. A controller may therefore protect its traffic by the proprietary application-layer scheme alone, or by that scheme within a TLS session. Both present an opaque payload to a passive observer, but by different mechanisms and with different metadata left exposed.

2. The firmware ruler problem

Determining which regime a given controller uses is, in practice, obstructed by a versioning problem that is easy to get wrong. The threshold at which a controller adopts TLS-based secure communication is expressed in firmware terms, but the S7-1200 line carries two separate firmware rulers that are not comparable.

The classic S7-1200 uses a V4.x ruler, and the commonly cited threshold for TLS-based programming and HMI communication on that line is firmware V4.5 or higher, paired with TIA Portal V17 or higher [5][6]. The second-generation S7-1200 G2, by contrast, begins its own numbering at V1.0.1 and is a distinct product line rather than an earlier point on the classic ruler [7]. A G2 controller reporting a low version within its own ruler is not an old, pre-secure device; it is a current device on a separate scale. Applying the classic V4.5 threshold to a G2 version number is a category error, and one that a defender inventorying a mixed estate can easily make.

Table 1 sets out the regimes against product generation, firmware ruler, and toolchain milestone.

Generation / line	Firmware ruler	Comms protocol	Encryption	Milestone
S7-300 / S7-400	legacy	S7comm (0x32)	None; cleartext, no authentication	1994; never updated [2]
S7-1200 V1 to V3	V1.x to V3.x	S7comm / early S7CommPlus	None; V3 confirmed cleartext	pre-2014 [2]
S7-1200 classic V4.0+ / S7-1500	V4.0+	S7CommPlus (0x72)	Payload encryption and integrity; legacy global-key protection*	~2014, TIA V13 [1][3]
S7-1200 classic >= V4.5 / S7-1500 >= V2.9	V4.5+ / V2.9+	S7CommPlus over TLS	TLS, certificate-managed	TIA V17, 2021 [5]
S7-1200 G2	V1.0.1 (04/2025) to V4.1.4 (04/2026)	S7CommPlus	Secure communication native; TLS-based, security by default†	G2 launch 04/2025; TIA V21 [7]

Table 1. S7 communication security across product generations and toolchain versions.

* The legacy protection of pre-TLS generations used a private key shared across the CPU family, later shown vulnerable to offline extraction (CVE-2022-38465) and replaced by TLS certificate management from TIA V17 [3][4].

† The G2 uses its own firmware ruler beginning at V1.0.1, distinct from the classic V4.x line; secure PG/PC and HMI communication is listed as a native CPU feature [7].

The G2 System Manual lists secure PG/PC and HMI communication, protection of confidential configuration data, and secure boot among the controller's native security features, configured in the engineering software [7]. The generation therefore ships as a security-by-default platform: the secure regime is the configured norm rather than an add-on enabled at a later firmware level.

3. What the installed base actually runs

The move to encryption by default describes the current product and the current toolchain. It does not describe the installed base, and the gap between the two is the substance of this briefing.

Three observations, drawn from public measurement, establish the shape of the field. First, the older families that carry no encryption remain widely deployed and widely reachable. Internet-exposure scanning consistently finds the S7-300 and the S7-1200 among the most common S7 controllers, and the S7-300 in particular never received the S7CommPlus security update: it communicates in cleartext by architecture, not by misconfiguration [2]. Second, S7 exposure is growing rather than receding. A longitudinal study of internet-exposed operational technology reported Siemens S7 among the fastest-growing exposed services between 2017 and 2024, with an increase on the order of 120 per cent over the period, against a total exposed-device population approaching 110,000 as of early 2024 [8]. Third, the S7 family as a whole holds a substantial share of the global PLC market, so the absolute number of deployed controllers behind these proportions is large [9].

Two qualifications keep this honest. Internet-exposed devices are a biased sample: they are not the whole installed base, and a controller reachable from the public internet is already a security failure independent of its protocol. Exposure figures are also inflated by honeypots, which recent scanning work has found to constitute a material and growing fraction of apparent industrial control hosts [10]. Neither qualification changes the direction of the finding. Whether one counts exposed devices, market share, or the simple fact that decades of S7-300 and S7-400 installations remain in service, the population is dominated by generations whose communication is readable. Encryption is the recent state of the art, not the prevailing state of the field.

4. Why the gap persists

The gap is not merely a matter of old hardware waiting to be replaced. It persists because closing it is expensive, disruptive, and often outside the asset owner's control.

Adopting the secure regime is not a firmware update in isolation. It requires updating the engineering project, provisioning certificates, disabling the legacy communication path, and loading the changed configuration into every affected device. That change breaks existing connections: HMIs, peer controllers, and other periphery configured against the old path must be updated in step [11]. Where the asset owner does not hold the engineering project, only the original machine builder can perform the migration, and a machine builder has little incentive to alter a system that is running to specification [11]. The practical outcome, reported from the field, is that brownfield installations remain on the readable path because migrating them is costly and risky, not because the operator is unaware of the exposure.

The history of the legacy protection reinforces the point. The confidentiality that pre-TLS controllers did offer rested on the family-wide global key later broken as CVE-2022-38465 [3]. In responding, the guidance from CISA was to confine legacy, non-TLS communication to trusted network environments, an instruction that implicitly concedes the legacy path is still in operational use and will remain so [4]. Security by default addresses the controller a plant buys today. It does not, on its own, address the controller a plant bought fifteen years ago and still runs.

5. Security implications for passive monitoring

The regime map has a direct consequence for any approach that relies on passively observing S7 traffic from a mirror or tap, without participating in the exchange.

For the readable regime, which dominates the installed base, passive observation recovers the substance of the communication: the operations, the memory areas addressed, and the values exchanged [1]. The protocol declares data types on the wire, so a passive observer recovers not only values but their types,

placing readable S7 closer to a protocol such as OPC UA in signed mode than to a type-agnostic protocol such as Modbus. For the encrypted regimes, whether protected by the S7CommPlus application-layer scheme or wrapped in TLS, passive observation of the payload recovers nothing of the application content. What remains visible is transport metadata: endpoints, timing, message sizes, and, depending on the transport version, portions of the connection-establishment exchange [5].

Two points bound this correctly. First, the loss under encryption is a loss of vantage, not a loss of logic. The evaluation an observer might perform on readable traffic is unchanged in principle; what encryption removes is the observer's ability to see the traffic at all. The same process logic that is legible on an S7-300 becomes opaque on a current G2 not because the logic changed but because the vantage was withdrawn. Second, the opacity holds specifically for the passive, unkeyed observer. The published breaks of S7CommPlus are the work of active adversaries: rogue clients that participate in the session, and reverse-engineering of the protocol against a controlled device [2]. None of them is a passive recovery from mirrored traffic without keys. An honest account must not conflate what an active attacker with a foothold can achieve with what a passive observer on a span port can see; the two are different problems, and only the latter is at issue here.

The net position is therefore mixed rather than bleak. Across most of the deployed S7 estate today, passive observation remains viable because the traffic is readable. Where encryption is in force, it marks the boundary of the passive vantage, and that boundary is a property of the deployment's transport security, not of the observing method.

6. Outlook

The direction of travel is clear even if its pace is not. New controllers ship in the secure regime by default, and the encrypted share of the S7 estate will grow as hardware is replaced and as toolchains advance. That growth is gradual: the economics described above mean brownfield installations turn over slowly, and the readable regime will remain a substantial fraction of the field for years rather than months.

Two developments bear watching. The first is the steady tightening of transport security on the controllers themselves, which narrows what even the connection-establishment exchange exposes as newer TLS versions encrypt more of the handshake [5]. The second, further out, is the interaction between these embedded TLS stacks and post-quantum cryptography. Current controller TLS implementations are rigid about what they will accept, to the point of rejecting oversized handshakes that advertise post-quantum key-exchange groups [6]. Whatever the merits of that rigidity, it signals that the cryptographic agility of deployed industrial controllers lags the general internet, and that the transition to post-quantum transport security in operational technology will be correspondingly slower. For an observer, each such tightening moves more of the exchange out of view; for a defender, it is a reminder that the security posture of the transport is now a first-class variable in what any monitoring approach can achieve.

References

[1] Wireshark. S7comm dissector and protocol documentation. Wireshark Foundation, accessed 2026.

[2] M. Rodda and V. Mavroudis. Analysis of Publicly Accessible Operational Technology and Associated Risks. arXiv:2508.02375, 2025.

[3] Claroty Team82. The Race to Native Code Execution in PLCs: Using RCE to Uncover Siemens SIMATIC S7-1200/1500 Hardcoded Cryptographic Keys. Claroty, 2022.

[4] Cybersecurity and Infrastructure Security Agency. ICS Advisory ICSA-22-286-04: Siemens SIMATIC S7-1200 and S7-1500 CPU Families (CVE-2022-38465). CISA, 2022.

[5] Siemens. Configuration of TLS-based PG/HMI Communication and the Protection of Confidential Configuration Data. Security Concept, TIA Portal V17, Item-ID 109798583, 2021; and Communication with S7-1500 and S7-1200, WinCC Unified Runtime, TIA Portal V21, 2026.

[6] python-snap7. Connecting to PLCs: TLS 1.3, firmware thresholds, and post-quantum key-exchange rejection on S7-1200/1500. Project documentation, v3.0, 2026.

[7] Siemens. SIMATIC S7-1200 G2 Programmable Logic Controller System Manual. V1.0.1, 04/2025, A5E52923928-AB.

[8] Forescout Research Labs. The Global Threat Evolution of Internet-Exposed OT/ICS. Forescout, 2024.

[9] Composite of analyst coverage of the PLC market and academic estimates of S7 market share, 2016-2026. Reported qualitatively; precise share figures diverge between sources.

[10] M. Mladenov, L. Erdodi, and G. Smaragdakis. All that Glitters is not Gold: Uncovering Exposed Industrial Control Systems and Honeypots in the Wild. IEEE European Symposium on Security and Privacy (EuroS&P), 2025.

[11] ENLYZE. On the Hype Around the Critical Siemens S7-1200/S7-1500 Vulnerability CVE-2022-38465. ENLYZE engineering blog, accessed 2026.