§ IAB WORKSHOP ON ACCELERATING THE DEPLOYMENT OF POST-QUANTUM AUTHENTICATION

What must a resumed connection re-check?

Post-quantum authentication leans harder on certificate expiry. Resumption is the path that never checks it.

One gap, eleven identifiers, eleven years and five months.
The specification recommends a time bound without quantifying it, and defines no server-side scope rule at all.
Seven parties responded, six by binding into the session and one at the HTTP layer; four more stopped resuming rather than answer.

León Acosta, QuantaKrypto · v1, 28 August 2026

§ THE PATTERN

A resumed session replays a verdict

A full handshake reaches a conclusion: this peer is who it claims, its certificate is valid, it is in scope for what it asked for, its trust anchors check out.

A resumed session carries that conclusion forward and replays it, **without re-establishing any of the conditions that produced it.**

Every row on the next slide is one condition somebody forgot to re-establish.

§ ELEVEN IDENTIFIERS · TWELVE ROWS · EIGHT PIECES OF SOFTWARE

One gap, published eleven times

DATE	IDENTIFIER	SOFTWARE	CONDITION NOT RE-ESTABLISHED	WHO GAINS
Sep 2014	CVE-2014-3616	nginx	session resumed in a server{} block other than the one that produced the verdict	needs position
May 2015	CVE-2015-3644	stunnel	a failed client-certificate verdict, routed on reuse as if it had succeeded	outside party
Aug 2016	CVE-2016-5419	curl	client certificate identity on a reused connection	the holder itself
Apr 2017	CVE-2017-7468	curl	the same bug: the fix was lost in 7.52.0 and restored here	the holder itself
Jun 2020	CVE-2020-8172	Node.js	server certificate validation, cached before verifying	needs position
Mar 2021	CVE-2021-22890	curl	resumed the origin session with a ticket issued by the HTTPS proxy	needs position
Feb 2022	CVE-2022-21654	Envoy	client-certificate validation settings, omitted from the session id context	outside party
Jan 2024	CVE-2024-0853	curl	REVOCATION: cached although OCSP verification failed	needs position
Feb 2025	CVE-2025-23419	nginx	client certificate scope across virtual hosts	outside party
Feb 2025	CVE-2025-23419	Cloudflare	client certificate scope across zones (same identifier)	outside party
Jul 2025	CVE-2025-23048	Apache httpd	per-virtual-host trust store for client auth	outside party
Feb 2026	CVE-2025-68121	Go crypto/tls	ClientCAs mutated between handshakes while ticket keys are shared	outside party

Five identifiers, six rows, need no network position. All five are client-certificate verdicts.

§ THE 2025 CLUSTER IS NOT THREE COINCIDENCES

One paper, one reporter, different causes

Hebrok, Storm, Cramer, Radoy and Somorovsky, Paderborn University. *"STEK Sharing is Not Caring: Bypassing TLS Authentication in Web Servers using Session Tickets"*, 34th USENIX Security Symposium, August 2025.

Four servers, client auth bypass

"all four implementations – Apache, nginx, (Open)LiteSpeed, and Caddy – were vulnerable to client authentication bypasses"

Six provider clusters, server auth bypass

Including Fastly and DDoS-Guard. Fastly was reported separately in September 2023 and fixed it by binding tickets to the issuing certificate.

Two identifiers came out of it, CVE-2025-23048 and CVE-2025-23419. Same class and same reporter, but not one cause: Apache fails on omitted SNI with a default host, Caddy on SNI=R, nginx ignores SNI. Cloudflare was reported via bug bounty.

§ MEASURED 25 TO 26 AUGUST 2026: CLIENT SIDE ON THE 25TH, SERVER SIDE ON THE 26TH

A server accepting an expired client certificate

TLS 1.3 server requiring and enforcing client certificates. One variable: a client certificate issued valid for 45 seconds. Same trust store throughout.

#	CLIENT CERTIFICATE	HANDSHAKE	SERVER
1	valid	full	accepted
2	EXPIRED	Reused	ACCEPTED
3	EXPIRED	full (control)	rejected: certificate has expired

Two verification sequences in the server log, for three connections. On the resumed one the server never ran certificate verification at all, and never asked for a certificate. The identity came out of the session.

The mirror experiment, a client and an expired *server* certificate, gives the same answer. Firefox does not re-verify: that is measured. Chrome does, on its source rather than on our bench, because measuring it would have meant adding a CA to the system trust store.

§ WHY THIS BELONGS IN THIS ROOM

Post-quantum leans on the check resumption skips.

Merkle Tree Certificates

MTC replaces only the signature part of path validation: "The relying party MUST continue to perform other checks, such as checking expiry" (WG draft -05, 7.2). Its size case assumes a 7-day lifetime. **Resumption is the path where that MUST is not performed.**

AuthKEM

KEM authentication "does not use the CertificateVerify" message. A bound on *CertificateVerify age*, which paraphrases RFC 9846's wording, has no referent. Define it as time since the last **full authentication**.

Scheduling, not size

Go has two open issues right now on what a ticket carries (77379) and whether chains are re-checked on resumption (77217). BoringSSL ships a knob to cut client certs from the session to save "ticket size". OpenSSL grew stateful tickets. **This code is being rewritten across the ecosystem anyway.** Cheapest moment to agree a definition; expensive one to skip.

The shorter certificates get and the bigger signatures get, the more the design leans on a check that the resumption path does not make.

§ WHERE THE EXPOSURE ACTUALLY IS

Server certificates need an attacker on the path. Client certificates do not.

Server certificate

If a domain changes hands or a CDN relationship ends, the client resolves DNS and reaches the new operator, who cannot decrypt the old ticket. The former operator benefits only while it still receives your traffic. That requires continued network position.

Client certificate, mutual TLS

The revoked party simply keeps connecting. It already holds the resumption secret. No network position, no interception, no cooperation from anyone.

A partition, not a count: in all five identifiers where an outside party gains standing access with no network position, the replayed verdict is a client-certificate verdict. No exceptions. The 2016-17 curl rows are the predicted other half, a client confusing its own identities, where the party harmed is the certificate holder.

Envoy binds scope and CRL contents into the session id context; it learned that from CVE-2022-21654. What survives is **notAfter**. Istio: 24-hour workload certificate, no CRL by default, and the window is bounded at **twelve hours by accident** - the ticket key dies with the SSL_CTX Envoy rebuilds on every certificate rotation.

§ SPECIFICATION, THEN PRACTICE

Recommended for time. Normative for scope, on one side only.

WHO	SESSION-AGE CLAMP	RE-ESTABLISHED CONDITION
nginx	yes, Oct 2022	not at the time
BoringSSL	**yes**	**re-verify: clients only. Servers get the session id context**
Go crypto/tls	**no**	**chain Jan 2026, roots Feb 2026**
HAProxy, Aug 2026	depends which library you built against	
Envoy (Istio, Consul)	**no**	**trust anchors, CRL contents, SAN matchers, SNIs**
Fastly	-	**ticket bound to the issuing certificate**
Apache httpd	**no**	**fixed in 2.4.64, at the HTTP layer**
OpenSSL	**no**	**no**
Caddy, Cloudflare, Google CLB, AWS ALB	stopped resuming under client auth	

One product, three answers depending on the library it is linked against, and the third only in a development snapshot rather than any stable release. That is not vendors disagreeing. That is an undefined requirement.

§ THE ASK

One rule, on the decision.

An endpoint MUST NOT rely on a replayed authentication verdict for any decision that a full authentication, performed when the verdict is first relied upon, would decide differently.

1. **Time.** Every validity window the original check evaluated, on every certificate in the path.
2. **Scope.** Identity *and* policy: required, optional, or not requested.
3. **Continued acceptance.** Narrowing trust changes invalidate, once per change rather than per resumption, and never past the status information's own nextUpdate.

Discharge it any way you like: re-establish, refuse to resume, defer the decision (Apache's 421), or over-approximate by hashing the inputs into the scope *paired* with a lifetime cap. Four conforming ways, and nothing today says which. Turn the rule on the survey and none of the six session bindings fully conforms, Envoy and HAProxy included.

Condition idea: Hebrok et al. Lifetime idea: Chuat et al. The rule, its conditions and the amortisation boundary: this paper.

§ SCOPE, AND WHY RAISE IT IN THIS ROOM

A specification gap, in the present tense

Today's specification names the quantity a resumed session must respect and recommends a bound without quantifying it. On scope it binds the client and asks the server for nothing. That is not a gap post-quantum creates. It is one post-quantum inherits, on the code path every implementation is about to rewrite for new signature sizes.

What today's specification cannot solve: six different answers and no text saying which conform, four parties who concluded the safe answer was to stop resuming, and a bug rediscovered over eleven years and five months because nothing says what a resumed session must re-establish.

The exposure is service meshes, mutual TLS, mobile backends and embedded clients, which do not partition, restart or forget. Browsers are the weaker case, conceded in the paper. For the report's open questions: what must a resumed session re-establish, on which side does the duty sit, and how is it defined so that composite certificates, Merkle Tree Certificates and KEM-based authentication can all answer it?