

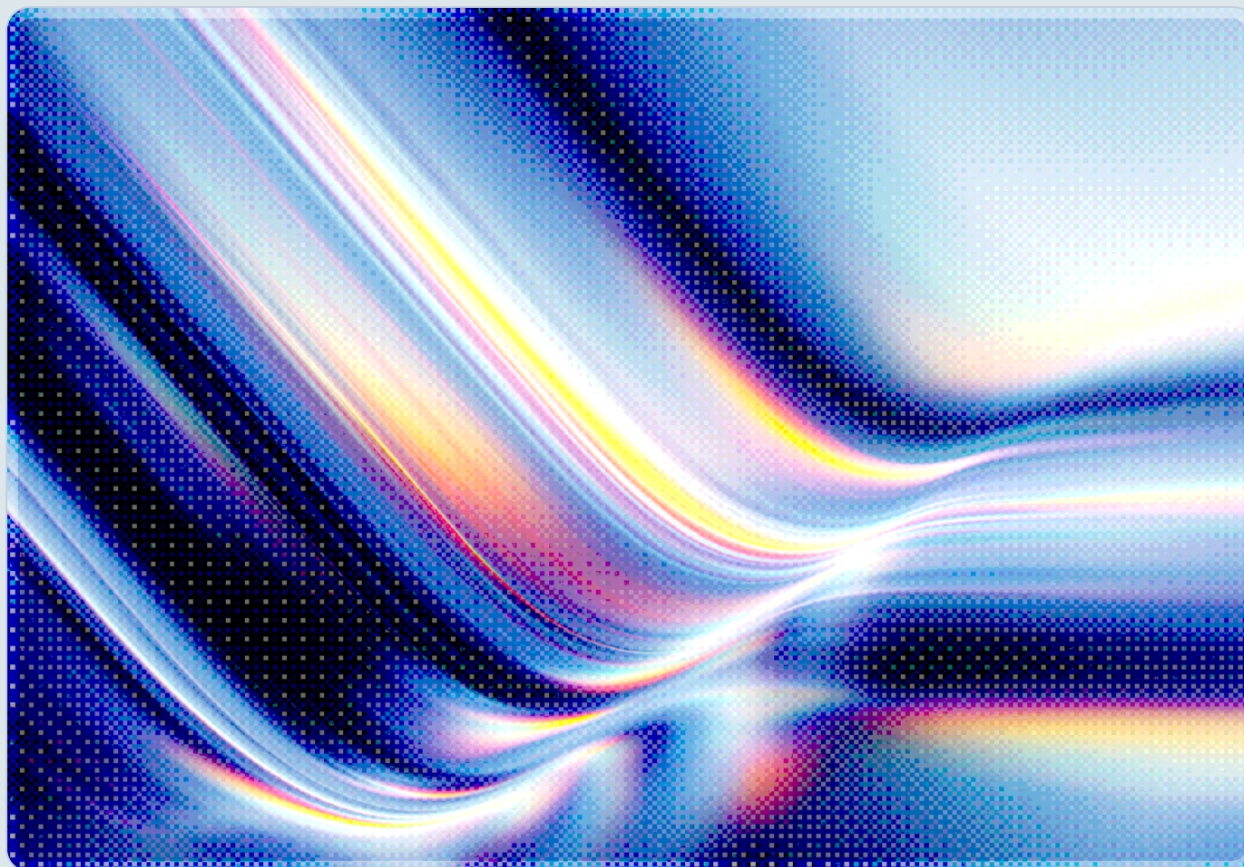
ccTLD Registry Hijacks and Rogue Google Certificates

Trust-Chain Exposure from DNS Operators You Do Not Control

2026-10-10



AI-assisted Rapid Research



CSAI

cloud
CSA security
alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

This document was generated with AI assistance and has not undergone official CSA review and approval processes.

Key Takeaways

Google disclosed on October 6, 2026 that attackers compromised third-party registries for three country-code top-level domains (.gh, .sl and .as), altered authoritative DNS records, and obtained unauthorized HTTPS certificates for Google properties and for other organizations [1][2]. Google says it has no reason to believe the issuing CAs acted improperly: the certificates passed ordinary domain control validation because the attackers controlled the DNS the CAs consulted [1][3]. Chrome blocked the certificates through CRLSets, but Google cautions that browser-side blocking should not be the only defense, since other clients depend on CA revocation and on the domain owner noticing the issuance [1][3].

The exposure is not limited to Google. Any organization that registers regional ccTLD variants of its brand, including parked ones, inherits the security posture of each registry operator [1]. The practical defenses are Certificate Transparency (CT) monitoring across the full domain portfolio and restrictive CAA records bound to specific ACME accounts and validation methods, although CAA is itself served from the zone an attacker may control, so it narrows the risk without eliminating it [1].

Background

Public web trust rests on a chain with a rarely scrutinized link at its start. A certificate authority issues a TLS certificate after confirming that the applicant controls the domain, and the most common confirmation methods are automated checks that rely on DNS: the applicant publishes a random token in a DNS record, or answers a challenge served from an address the DNS names resolve to. The CA's verdict therefore depends on the integrity of the DNS hierarchy above the domain, and at the top of that hierarchy sit the registry operators that run each top-level domain. Whoever controls a registry's delegation data or authoritative servers can decide what every resolver, including a CA's validation resolver, sees beneath it.

On October 6, 2026, Google's Chrome Secure Web and Networking Team published its response to what it described as recent hijacks of the .gh (Ghana), .sl (Sierra Leone) and .as (American Samoa) registries [1]. According to Google, the attackers compromised third-party registry operators, modified authoritative DNS records, and used that control to obtain unauthorized HTTPS certificates for several

Google domains. Certificate Transparency logs then showed that other organizations had been affected, among them several large global brands and widely used online services, whom Google says it contacted where possible [1][4]. Google stated that its own systems were not compromised [1][3].

Reporting that draws on the same disclosure adds detail that Google's post, as retrieved for this note, does not state itself. The Hacker News reports hijack dates of September 22 for .gh, September 25 for .sl and September 27 for .as, and counts twelve certificates covering seven Google and YouTube hostnames, eleven issued by Let's Encrypt and one by ZeroSSL [2]. The hostnames it lists include google.com.gh, youtube.com.gh, google.sl, google.com.sl, youtube.sl, google.as and youtube.as [2]. Those specifics rest on a single secondary source and should be treated as reported by a single outlet rather than independently confirmed. Public reporting identifies no threat actor, does not explain how the registries were compromised, and does not report the registries' own remediation status [2][3][4].

Security Analysis

Why the CAs did not stop it

The main lesson of the incident is that the issuance was, from the CA's perspective, procedurally valid. Domain control validation asks whether the party requesting a certificate can demonstrate control of the DNS for a name. An attacker who has rewritten the delegation or authoritative records at the registry level can answer that question truthfully, because the attacker does control what the DNS says. Google's statement that it has no reason to believe the issuing CAs acted improperly is consistent with this reading [1][3]. The failure sits above the CA, in the registry layer that both the CA and the domain owner are forced to trust.

This explains why defenses built to detect network-level interference had limited purchase here. Multi-perspective issuance corroboration (MPIC), adopted in the CA/Browser Forum Baseline Requirements through Ballot SC-067, has CAs perform domain validation and CAA checks from several network vantage points so that a localized routing attack is harder to mount [5]. A registry compromise differs in kind. The forged answers originate in the authoritative data itself, so every vantage point retrieves the same altered records. None of the sources reviewed for this note discuss MPIC in connection with the incident, so the point is an inference from how the mechanism works, not a reported finding [1][2][3][4].

The portfolio exposure problem

Organizations commonly register country-code variants of their primary brand to prevent squatting, serve regional audiences, or simply hold the names. These domains are often parked, rarely monitored, and absent from the inventories that security teams use for certificate management. The incident shows the consequence. A rogue certificate for google.as or youtube.com.gh gives an attacker a valid-looking identity for a hostname that may carry no legitimate traffic, which is likely to make misuse harder to notice and, for phishing or traffic interception, no less effective. Google's recommendation that monitoring cover the "entire domain portfolio, including parked or regional ccTLD properties" addresses this directly [1].

The dependency is also structural. A domain owner can harden its own DNS provider, enable multi-factor authentication on registrar accounts, and still have no influence over the registry that holds the delegation for its ccTLD name. The domain owner's exposure is therefore bounded by the registry-layer trust of the weakest operator in its portfolio, even though registrar and DNS-provider controls, registry lock and DNSSEC also matter. Registry operators for smaller country-code domains vary widely in size and resources, and a single brand's regional portfolio may span dozens of them. This is an assessment of structure, not a claim about the security of any particular operator, and no public information in the reviewed sources characterizes the compromised registries' controls [1][2][3][4].

What detection and containment looked like

Two mechanisms limited the damage. First, Certificate Transparency made the rogue issuance visible after the fact. Google used CT logs to identify other affected organizations, and the company states that monitoring CT provides "a near real-time alert whenever a certificate is issued for your domains" [1]. Second, Chrome blocked the certificates through CRLSets, its mechanism for emergency distrust, and Google worked with the issuing CAs to have the certificates revoked so that clients other than Chrome would benefit [1].

Both mechanisms are partial. CT is a detection control, not a prevention control: it tells the domain owner that a certificate exists, ideally within minutes, but it does not stop the certificate from validating in clients that do not enforce CT or have not received a revocation. CRLSets protect Chrome users only, and Google explicitly advises that browser-side intervention should not be relied upon to protect users [3]. Organizations that do not monitor CT, and whose domains are not on Google's radar, receive neither the proactive Chrome block nor a notification. Where Google could not contact an affected party, the owner's only protection was its own monitoring [1][4].

Controls that narrow the window

Google's guidance centers on CAA records. A CAA record, standardized in RFC 8659, lets a domain owner list the CAs permitted to issue for the name, and RFC 8657 extends the record with `accounturi` and `validationmethods` parameters that bind issuance to a particular ACME account and to specific validation methods [6][7]. Google recommends publishing restrictive CAA records with account bindings [1]. The control has a limit: CAA records live in DNS, and an attacker who controls the authoritative DNS for the name can also change or remove them. CAA therefore reduces the risk from a compromised CA account or an opportunistic request, but it is not a complete answer to registry-level compromise on its own. The same caveat applies to any control that relies solely on records served from the compromised zone.

Google's post also points to longer-term ecosystem changes, including reducing certificate validity and the reuse of domain control validation results [4]. The CA/Browser Forum's Ballot SC-081 reduces maximum certificate validity from 398 days to 200 days, a step that took effect on March 15, 2026, with further reductions scheduled to 100 days in March 2027 and 47 days in March 2029; validation reuse periods follow a shrinking schedule of their own [8]. Shorter lifetimes and shorter reuse windows do not prevent a rogue issuance, but they limit how long an attacker can profit from validation obtained during a brief registry compromise, and they reduce dependence on revocation. The sources reviewed do not discuss DNSSEC or registry lock in connection with this incident [1][4], so this note does not assess their effect on it.

Recommendations

Immediate Actions

Organizations should inventory every domain they hold, with particular attention to ccTLD variants, parked names and domains registered by regional teams or acquired companies, and confirm that each appears in a CT monitoring feed. CT monitoring should alert on any certificate issued for the portfolio from a CA or account the organization does not use. Security teams should review CT history for any regional name under .gh, .sl or .as, starting from a conservative date before September 22, 2026, because that window rests on single-source reporting and may be wider if the dates are inaccurate. The same review applies to any other ccTLD where a registry-level incident is later disclosed. As CSA guidance, an organization that finds an unauthorized certificate should file a certificate problem report with the issuing CA and request revocation without waiting for browser action.

Short-Term Mitigations

Publish CAA records on all domains, including parked ones, restricted to the CAs the organization actually uses and, where the CA supports it, bound to specific ACME accounts and validation methods [1][6][7]. Document an escalation path for rogue-certificate events that spans security operations, the legal or brand protection team, and the registrar relationship, since containment may require contacting the registry through the registrar. Where regional domains serve no business purpose, consider whether to retain them, weighing the squatting risk against the exposure that an unmonitored delegation creates. Organizations should also verify that registrar accounts use multi-factor authentication and that registrar-level change notifications are enabled, while recognizing that these controls do not extend to the registry.

Strategic Considerations

Third-party DNS operators belong in the same dependency map as cloud providers and software suppliers. Governance teams should record, for each business-critical domain, which registry and registrar hold its delegation and what that operator has disclosed about its security posture. Where a brand has the option of consolidating onto domains administered by registries that offer registry-lock services and publish their incident response practices, that option merits evaluation as a risk-reduction measure. Certificate lifecycle automation should be designed on the assumption that validity periods and validation reuse windows will continue to shrink [8], which favors investment in automated issuance and in monitoring that keeps pace with it. Finally, incident playbooks should assume that CT, CAA and browser blocklists are layered detective and partial preventive controls, and that none of them substitutes for assurance about the DNS hierarchy.

CSA Resource Alignment

CSA's [Integrating DNS and SDP: Enhanced Zero Trust Policy Enforcement 2.0](#) [9] treats DNS as a control point for identity-aware policy enforcement. That guidance concerns resolution paths inside an enterprise rather than registry integrity, but it supports a shared premise with this incident: DNS answers are a trust input, and architectures that accept them without verification inherit the integrity of every operator behind them. Teams applying its pre-resolution enforcement model should also consider how they would detect a delegation change upstream of their own resolvers.

CSA's [Cloud Namespace Hijacking: When Deleted Buckets Become Backdoors](#) [10] analyzes a different layer of the same pattern, in which control of a globally unique name confers the trust that other systems attach to it. The ccTLD events and bucket-name hijacks share a defensive logic: inventory the names you

depend on, monitor for changes in who controls them, and avoid treating a name as proof of identity.

Because the incident exposes a third-party dependency and a certificate lifecycle weakness, the AI Controls Matrix v1.1 ([AICM](#)) [11] is relevant as a baseline for supply chain and cryptography and key management control objectives, and it is a superset of the Cloud Controls Matrix. Organizations running AI services that authenticate to model providers or tool endpoints by hostname should apply the same domain-portfolio and CT monitoring discipline to those endpoints.

References

- [1] Google Chrome Secure Web and Networking Team. "[Chrome's response to recent ccTLD registry hijacks](#)." Google Security Blog, October 6, 2026.
- [2] The Hacker News. "[Attackers Hijack .gh, .sl, and .as Registries to Obtain Certificates for Google Domains](#)." The Hacker News, October 2026.
- [3] The Register. "[Attackers hijacked top-level domains, minted fake security certs for Google and other orgs](#)." The Register, October 7, 2026.
- [4] Help Net Security. "[Hackers hijack three country-code domain registries, obtain HTTPS certificates for Google domains](#)." Help Net Security, October 7, 2026.
- [5] CA/Browser Forum. "[Ballot SC-067v3: Require Domain Validation and CAA Checks to be Performed from Multiple Network Perspectives Corroboration](#)." CA/Browser Forum, August 2024.
- [6] IETF. "[RFC 8659: DNS Certification Authority Authorization \(CAA\) Resource Record](#)." IETF, November 2019.
- [7] IETF. "[RFC 8657: CAA Record Extensions for Account URI and ACME Method Binding](#)." IETF, November 2019.
- [8] CA/Browser Forum. "[Ballot SC-081v3: Introduce Schedule of Reducing Validity and Data Reuse Periods](#)." CA/Browser Forum, April 2025.
- [9] Cloud Security Alliance. "[Integrating DNS and SDP: Enhanced Zero Trust Policy Enforcement 2.0](#)." CSA Zero Trust Working Group, 2026.
- [10] Cloud Security Alliance. "[Cloud Namespace Hijacking: When Deleted Buckets Become Backdoors](#)." CSA, June 29, 2026.
- [11] Cloud Security Alliance. "[AI Controls Matrix v1.1](#)." CSA, 2026.