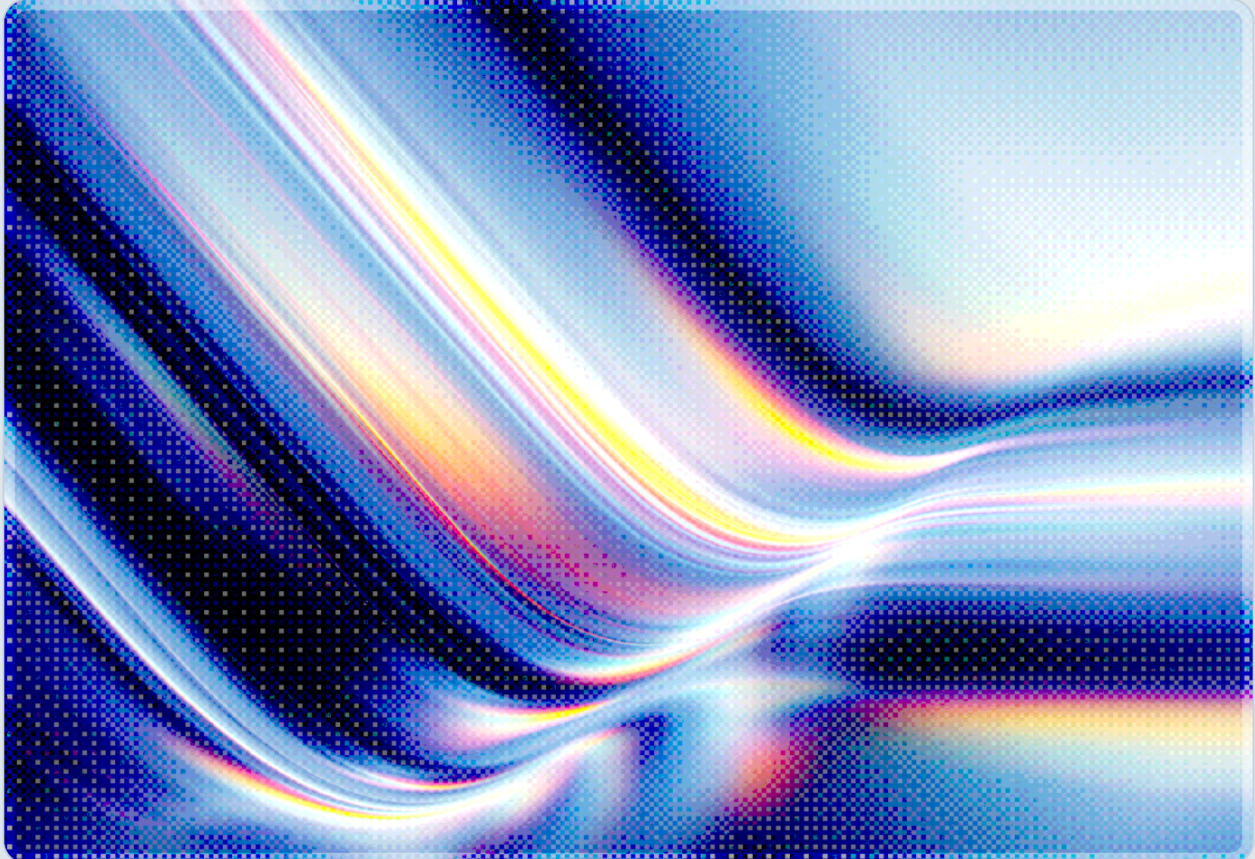


Registries as the Weak Link: ccTLD Hijacks and Certificate Trust

Security Implications and Guidance on the .gh, .sl, and .as Incidents

2026-10-08

 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

Press reporting dates the compromise of the country-code registries for Ghana (.gh), Sierra Leone (.sl), and American Samoa (.as) to September 22 through 27, 2026. In each case attackers modified authoritative DNS records and obtained unauthorized HTTPS certificates for Google properties and for other, unnamed organizations [1][2]. Google's account does not identify any CA error. Because domain control validation (DCV) succeeded against attacker-controlled DNS answers, the failure appears to lie in the registry layer rather than in CA issuance practice [1][3].

Google stated that browser-side intervention "should not be relied on" to protect users and that it cannot guarantee its analysis identified every affected domain, so the responsibility for detection falls to domain owners [1][2]. Certificate Transparency (CT) monitoring and restrictive CAA records are among the controls Google recommends, and both are low-cost relative to most enterprise controls. Neither prevents a determined registry-level attacker from obtaining a certificate, so they are best treated as detection and constraint measures rather than prevention [1]. Organizations that operate under any ccTLD, and those that depend on third-party services under one, should map that dependency now, because a registry compromise bypasses controls at the domain, DNS provider, and CA level.

Background

On October 6, 2026, Google's Chrome security team published an account of three incidents in which attackers compromised the operators of the .gh, .sl, and .as country-code top-level domains and altered authoritative DNS records [1]. Google's statement was that the compromises put "any domain ending in .gh, .sl, or .as at risk" [1]. Reporting on the disclosure indicates the compromises occurred on September 22 (.gh), September 25 (.sl), and September 27 (.as) [2]. Google did not publish how the registry operators were breached, and public reporting to date does not describe the intrusion method or attribute the activity to a named actor [2][4].

Certificate Transparency logs showed at least 12 unauthorized certificates for Google and YouTube names under the three ccTLDs, such as google.com.gh, google.sl, and google.as. Eleven were issued by Let's Encrypt and one by ZeroSSL, according to reporting on the disclosure [2]. A Let's Encrypt representative confirmed that the certificates for Google and YouTube had been issued and revoked [2]. The .gh certificates and the ZeroSSL certificate were revoked on September 26, and the remaining .sl

and .as certificates on October 1, so no revocation precedes the compromise of the corresponding registry [2]. Google said CT data also pointed to other organizations it believes were affected by the same attacks, "including well-known global brands and widely used online services," but it did not name them [3]. Google further acknowledged that it "cannot guarantee that our analysis identified every affected domain" [4].

Chrome responded by blocking the identified certificates through CRLSets, its mechanism for pushing emergency revocations to browsers, and worked with the issuing CAs to revoke them [1][3]. Google also contacted affected organizations where it could identify them. Its guidance to the broader community was direct: Chrome users need to take no action, but browser-side intervention should not be relied on to protect users [1][2]. That distinction matters for security teams, because Chrome's block covers only the certificates Google found, and other browsers, applications, and API clients may not apply the same list.

The incidents are best read as a failure of an assumption rather than of a protocol. The web PKI allows a CA to issue a certificate when an applicant demonstrates control of a domain, commonly by placing a specified value in DNS or serving a token over HTTP [2]. That model treats the DNS hierarchy as a source of truth. When an attacker controls a registry, the attacker controls the authoritative answer to the question the CA asks, and a correctly operating CA will issue a valid certificate.

Security Analysis

A TLS certificate protects a user only if issuance is limited to the party that genuinely controls the name. Domain control validation links the two, and it depends on DNS. This section traces where that dependency fails when a registry is compromised, why country-code registries present a particular exposure, how the common controls fare against the attack pattern, and what the incidents reveal about detection latency.

Where the trust chain breaks

A registry operator sits at the top of the DNS dependency: it publishes the delegation, the name servers, and any DNSSEC material for every name beneath its zone. Compromise at that layer lets an attacker redirect the name servers for a target domain to infrastructure the attacker controls, answer the CA's validation query with the expected value, and receive a certificate that is indistinguishable from a legitimate one at the protocol level.

This has a consequence that differs from more familiar certificate incidents. In a misissuance caused by a CA flaw, the remedy is to fix the CA. Here, Google's account does not identify any CA error [1][3], so the affected controls are the ones that organizations rarely examine: registry account security, the governance of national registry operators, and the assumption that a domain under a country code is as well defended as one under a large commercial registry. Reporting indicates the attacker obtained valid certificates for names whose DNS it controlled, which would permit impersonation without a browser warning [4]. The authors found no public evidence that the certificates were used to intercept traffic, since the available evidence establishes issuance rather than use. The combination of DNS control and valid certificates is nonetheless what would make the technique useful for interception rather than mere defacement.

Why ccTLDs are exposed

Country-code registries can vary widely in size, funding, and operational maturity. Some may be run by national agencies or universities with small security teams, while others are commercial operations. The three incidents were reported within a single week, which suggests, though public evidence does not establish, a common technique or actor targeting registries as a class. Treating that as a hypothesis, defenders should assume other ccTLD operators could be targeted in the same way, and that the set of affected domains is determined by the registry's zone rather than by the victim's own hygiene.

Exposure also extends beyond organizations that choose a country-code domain for their primary site. Many enterprises depend on services, vendor portals, link shorteners, or regional properties under a ccTLD, and a SaaS provider's use of a hijacked ccTLD can place customers at risk without any change on the customer side. Google's note that other well-known brands and services appear in the CT data is consistent with this broad exposure, though the affected parties have not been named [3].

What the existing controls did and did not do

The following table summarizes how the main controls relate to this attack pattern. The assessments are the authors' analysis based on how each mechanism is designed, and the incident reporting does not state whether any particular control would have changed the outcome beyond the measures Google cites.

Control	What it does	Effect against registry-level compromise
Certificate Transparency monitoring	Alerts the domain owner when any	Detection only. CT monitoring can alert shortly after issuance, and Google recommends it [1].

Control	What it does	Effect against registry-level compromise
	certificate is logged for the domain	Value depends on how quickly someone acts on the alert.
CAA records	Declares which CAs may issue for the domain [1]	Limits the choice of CA, but an attacker who controls DNS can also change or remove the CAA records, so the control is weak against this threat when the CAA record lives in the compromised zone.
Multi-perspective validation	CA checks control from several network locations [5]	Defends against localized network interception, not against a change to the authoritative data itself, since all perspectives see the same forged answer.
Shorter certificate and DCV reuse lifetimes	Reduces how long a validated state or certificate remains usable [6]	Narrows the window for a stolen validation or a fraudulent certificate, with no effect on the first issuance.
Browser revocation (CRLSets)	Pushes emergency blocks to Chrome [1]	Reactive and limited to certificates already identified, and Google cautions against relying on it [1][2].

The pattern across the table is that most mechanisms either detect after issuance or constrain the attacker in ways a DNS-controlling attacker can often undo. DNSSEC is not discussed in Google's account [1], and its effect here depends on whether the registry's own signing keys or the delegation signer records were controlled by the attacker, which public reporting does not say. An attacker with registry-level control may be able to publish a consistent, validly signed set of records for the zone, so organizations should not assume DNSSEC would have prevented issuance.

Industry direction on validation lifetimes

Google tied its longer-term response to ballot SC-081v3 in the CA/Browser Forum, which reduces certificate validity and the period during which a CA may reuse a prior validation [1][6]. Reporting indicates the DCV reuse limit is currently 200 days, falling in stages to 10 days by March 2029 [2][6]. The same reporting states that Let's Encrypt currently reuses domain checks for 30 days and plans to cut that period to 7 hours by 2028 [2]. Shorter reuse windows raise the frequency at which a CA re-

validates control, which improves the chance that a hijack is reversed or detected before the next issuance. They do not stop an attacker who holds control at the moment of validation, as the September incidents show, and they increase the operational load on organizations that must automate renewal reliably.

Detection and response gaps for enterprises

For most organizations, the practical lesson concerns detection latency. The unauthorized certificates were issued over a period of days and revoked between September 26 and October 1, while public disclosure came on October 6 [1][2]. An organization subscribed to CT alerts for its domains would have seen issuance as it occurred, whereas an organization relying on vendor or browser action would have learned later, if at all. Because the CA/Browser Forum Baseline Requirements oblige a CA to investigate certificate problem reports within 24 hours [10], a prompt, well-formed report to the issuing CA is the fastest route to revocation outside the browser channel.

Recommendations

The recommendations below are ordered by the time an organization needs to act. Immediate actions establish visibility into exposure, short-term mitigations reduce the impact of a fraudulent certificate, and strategic considerations address the structural dependency on registries.

Immediate Actions

Inventory every domain your organization owns, operates, or depends on that sits under a ccTLD, with specific attention to .gh, .sl, and .as. Then review CT logs for those names for the period beginning September 22, 2026, and investigate any certificate you did not request. If you find one, file a certificate problem report with the issuing CA and treat the event as a potential traffic-interception incident [1][2].

Subscribe to CT monitoring for all production domains, including those under generic TLDs, so that unexpected issuance generates an alert to a team that is staffed to act on it. Publish restrictive CAA records that name only the CAs you use, and consider the account-binding and validation-method parameters that some CAs support to narrow what an attacker can request [1]. Confirm with your DNS and registrar providers that registry-lock or equivalent change-control features are enabled where offered.

Short-Term Mitigations

Review dependencies on third-party services, link domains, and partner properties that use country-code domains, and ask critical vendors how they monitor CT and secure their registrar accounts. Where an application authenticates a known counterparty, such as a mobile app calling its own API or a service-to-service connection, consider certificate or public-key pinning so that a fraudulent but publicly trusted certificate is rejected. Pinning carries operational costs and should be applied selectively. Add DNS monitoring that alerts on changes to NS and DS records for owned domains, so that a delegation change is visible independently of certificate issuance.

Prepare the automation needed for shorter certificate lifetimes. The industry schedule means renewal will become more frequent, and organizations that rely on manual processes will face rising outage risk [6]. Rehearse an incident playbook for fraudulent certificate issuance that includes CA contact paths, revocation verification, and a decision on whether to rotate credentials and tokens exposed to a possible man-in-the-middle.

Strategic Considerations

The incidents argue for treating the registry as part of the supply chain. In the authors' experience, third-party risk programs tend to focus on SaaS and cloud providers rather than TLD operators, although a registry sits above most controls that depend on DNS-based identity. Registry-lock availability, DNSSEC deployment, and published incident history are reasonable design inputs when choosing a TLD for business-critical names, and avoiding unnecessary reliance on small registries is worth considering, although the available evidence does not show that any particular registry type is safer.

At the ecosystem level, the open questions are whether registries should be held to security expectations comparable to those placed on CAs, and whether CAs should apply additional scrutiny when the validation path runs through a zone with recent signs of compromise. These are policy matters for ICANN, national authorities, and the CA/Browser Forum, and organizations can contribute by supporting disclosure, sharing indicators, and reporting anomalies promptly. Security leaders should also record that this class of attack defeats controls that have been widely relied upon, which supports a defense-in-depth posture that assumes any single trust anchor can fail.

CSA Resource Alignment

CSA's guidance on integrating Software-Defined Perimeter with DNS, [Integrating SDP and DNS: Enhanced Zero Trust Policy Enforcement](#) [7], is a relevant CSA publication on this topic. It treats DNS and DDI systems as an enforcement point within a zero trust architecture. The registry incidents illustrate the corollary: if DNS answers are treated as authoritative input to a trust decision, a compromise upstream of the organization's own resolvers undermines the decision, so policy should not depend on DNS alone.

CSA's [Using Asymmetric Cryptography to Help Achieve Zero Trust Objectives](#) [8] addresses the role of keys and certificates in establishing identity. The fraudulent certificates here were valid at the protocol level, which reinforces that certificate possession should be paired with additional identity signals such as mutual authentication, pinning for known counterparties, and device or workload attestation in high-value paths.

Finally, the [AI Controls Matrix \(AICM\) v1.1](#) [9] provides a control framework that includes cryptography and key management, infrastructure security, and logging and monitoring domains. Organizations can use it to confirm that certificate inventory, CT-based alerting, and third-party dependency review are covered. This is relevant to AI deployments in particular because agent and model-serving endpoints, as well as the APIs they call, depend on the same TLS trust assumptions that these incidents undermined.

References

- [1] Google. "[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] Help Net Security. "[Hackers hijack three country-code domain registries, obtain HTTPS certificates for Google domains](#)." Help Net Security, October 7, 2026.
- [4] The Register. "[Attackers hijacked top-level domains, minted fake security certs for Google and other orgs](#)." The Register, October 7, 2026.
- [5] Let's Encrypt. "[Multi-Perspective Validation Improves Domain Validation Security](#)." Let's Encrypt, February 19, 2020.
- [6] CA/Browser Forum. "[Ballot SC081v3: Introduce Schedule of Reducing Validity and Data Reuse Periods](#)." CA/Browser Forum, April 2025.
- [7] Cloud Security Alliance. "[Integrating SDP and DNS: Enhanced Zero Trust Policy Enforcement](#)." CSA, 2024.
- [8] Cloud Security Alliance. "[Using Asymmetric Cryptography to Help Achieve Zero Trust Objectives](#)." CSA, 2024.
- [9] Cloud Security Alliance. "[AI Controls Matrix \(AICM\) v1.1](#)." CSA.
- [10] CA/Browser Forum. "[Baseline Requirements for the Issuance and Management of Publicly-Trusted TLS Server Certificates](#)." CA/Browser Forum.