

L'état de la PKI publique en 2026

Trois modes de défaillance migrés, à partir d'un audit de 44,6 millions de certificats

ZetaCert Research — Mouhamadou Thioub, Research Lead Version 1.0 — Publication publique, 2026-10-11
Les identités des fabricants et opérateurs sont anonymisées dans cette édition publique. Tous les findings ont été divulgués aux parties concernées via des canaux de divulgation coordonnée (vulnerabilities@zetacert.com) avant publication.

Résumé exécutif

Entre février et avril 2026, ZetaCert Research a audité 44 642 253 certificats X.509 collectés à partir de l'écosystème public Certificate Transparency et de campagnes de scan TLS actif. Le corpus comprend 24,9 millions de certificats RSA et 19,8 millions de certificats ECC, couvrant l'ensemble des autorités de certification commerciales majeures, les émetteurs internes d'entreprise, ainsi qu'une longue traîne profonde de déploiements firmware embarqués et IoT.

Quatre détecteurs cryptanalytiques bien établis ont été appliqués à l'échelle du corpus complet : batch GCD sur 6,6 millions de modules RSA (Bernstein 2004 / Heninger et al. 2012), factorisation de Fermat sur 25,8 millions de modules (sweep premiers proches à profondeur $k=100$, complété par un sweep plus profond $k=2\,000\,000$ sur le sous-corpus IoT scanné), récupération de clés ECDSA par Hidden Number Problem / réduction de réseau sur 23 groupes (émetteur, courbe) et 1 436 signatures extraites en direct depuis des handshakes TLS, et un contrôle strict de mésémission CA pour les valeurs de Common Name pointant vers un espace de nommage non public. **Les quatre détecteurs ont produit des résultats négatifs propres.** Ce résultat est en lui-même notable : la PKI publique visible via CT, telle qu'opérée par les autorités de certification commerciales majeures en 2026, est implémentée de manière robuste contre les modes de défaillance classiques de RNG cassé et de mésémission de template.

Le risque cryptographique n'a pas disparu pour autant. Il a migré. Nous documentons trois classes de modes de défaillance qui *sont* présentes dans le corpus 2026, chacune représentant une génération différente d'erreur :

- **Mode A — Migration RSA vers la longue traîne IoT/firmware.** Le pattern classique de RNG cassé / keygen défaillant documenté par Heninger et al. (2012) puis par Nemec et al. (ROCA 2017) a migré hors de l'écosystème commercial des CA et s'est installé dans les valeurs par défaut firmware des fabricants. Nous identifions 47 findings distincts affectant des routeurs grand public, des équipements réseau d'entreprise, des appliances NAS, des CPE industriels et des contrôleurs de plan de gestion. Un cluster vendor (Fabricant C) s'étend sur un défaut SDK firmware de 12 ans (RSA-2048 avec $e=17$ au lieu du standard $e=65537$) ; un autre cluster vendor (Fabricant H) révèle un bug de génération de modules de 1040 bits encore observable en 2026.
- **Mode B — Concentration volontaire de clés ECDSA chez les opérateurs.** Pattern architectural précédemment non documenté : six opérateurs indépendants (registrars, fournisseurs de protection de marque, SaaS de génération de leads, plateformes de parking de domaines) réutilisent une seule clé privée ECDSA P-256 (ou P-384) comme SubjectPublicKeyInfo sur l'ensemble de leur flotte client. L'exposition cumulée est de 25 105 certificats concentrés sur 9 clés distinctes. Le plus grand cluster signe à lui seul 18 866 certificats avec une seule clé ; un autre signe les certificats d'un portefeuille brand-protection comprenant des clients du Fortune 500 (médias et divertissement, pharmaceutique, biens de consommation, agro-business). Un cluster de moindre taille couvre 8 hôtes SCADA dans un déploiement industriel unique. Un cluster séparé d'émission de masse, type « ghost cluster », couvre 1,65 million de certificats sur une seule clé — émission seule, jamais déployée — et constitue une observation méthodologique sur l'activité ACME en masse visible via CT, plutôt qu'un finding de compromission active.
- **Mode C — Défauts de conception sur la durée de vie des certificats.** Une troisième classe, ni involontaire (RNG cassé) ni volontaire-architecturale (réutilisation de clé opérateur), mais intégrée

à la conception au niveau du firmware. Nous identifions 775 certificats de plan de gestion sur ≈ 721 équipements distincts, provenant de 4 fabricants maintenus, dont le champ `notAfter` est codé en dur à 2038-01-19 03:14:08 UTC — la limite `INT_MAX` du `time_t` signé 32 bits. (S’y ajoutent 242 certificats mono-hôte qui partagent la même valeur d’expiration sans la structure de template flotte, ce qui porte le total des certificats à validité `INT_MAX` dans le corpus à 1 017.) La population fabricant affectée couvre des pare-feu, des routeurs grand public (28 modèles distincts pour un seul fabricant, Fabricant A), des appliances de sécurité et des caméras IP. À la frontière, l’ensemble des certificats de gestion de la flotte expire simultanément, sans chemin de renouvellement gracieux prévu dans le firmware. Nous identifions également deux certificats provenant d’un fabricant majeur de plateformes serveurs (Fabricant L) dont le champ `notAfter` est *antérieur au* 1970-01-01 (overflow de date dans le sens inverse), ainsi qu’un certificat de gestion auto-signé d’un pare-feu (Fabricant N) qui usurpe l’identité d’une CA commerciale retirée dans son DN d’émetteur — les deux pointant vers la même cause racine : des défauts de manipulation du temps et de construction de template dans le code de génération de certificats firmware.

Leçon structurelle commune aux trois modes : les primitives cryptographiques ne sont plus l’endroit où la PKI moderne se brise. La cassure s’est déplacée vers le haut de la stack — vers l’entropie au moment de l’installation firmware (Mode A), vers l’architecture opérateur (Mode B), et vers le code de gestion du temps et de construction de template (Mode C). Pour un RSSI, l’implication opérationnelle est qu’un scan TLS périmétrique réussi n’est plus une preuve que l’infrastructure de certificats est saine. Les défauts firmware fabricant, les clés opérateurs partagées en amont du contrat de service, et les erreurs de calcul temporel à la conception sont tous invisibles à un scanner de suites cipher ou de vulnérabilités standard ; ils nécessitent une inspection sémantique du contenu des certificats.

Ce document présente la méthodologie, les findings, le spectre des réponses obtenues lors de la divulgation coordonnée auprès des fabricants concernés (de « remédiation planifiée » à « bonne pratique reconnue » jusqu’à « Fin de vie — pas de remédiation possible »), et une checklist d’audit concrète pour les RSSI et les équipes achats.

1. Introduction

La PKI publique du Web en 2026 se situe entre deux équilibres. Au-dessus, un petit nombre de CA commerciales à fort volume (Let’s Encrypt, Google Trust Services, DigiCert, Sectigo, les estates d’émission Microsoft et Amazon) ont professionnalisé leur génération de clés, leur génération de nonces et leur gestion de templates à un niveau tel que les attaques cryptanalytiques classiques à grande échelle n’ont plus de prise sur les certificats qu’elles émettent. En dessous, la longue traîne des certificats — ceux générés à l’intérieur des firmwares, ceux émis par les pipelines d’automatisation côté registrar, ceux liés à des appliances de plan de gestion qui reçoivent des mises à jour rares — n’a pas professionnalisé au même rythme. Le risque cryptographique a migré vers le bas et vers le côté, dans cette longue traîne.

Ce document rapporte les résultats d’une campagne d’audit de quatre mois (février-avril 2026) ciblant les deux équilibres. Nous avons collecté 33,6 millions de certificats à partir des logs CT publics (Google Xenon et Argon, Cloudflare Nimbus, DigiCert Wyvern et Sphinx, Sectigo Elephant et Tiger, TrustAsia 2026a/b) et complété par 11 millions de certificats obtenus via probing TLS actif sur 24 ports IoT-pertinents à travers l’espace public IPv4, à un rythme courtois de 10–20 connexions par seconde par port. Le corpus combiné représente 44,6 millions de certificats sur 24,9 millions de modules RSA uniques et un nombre approximativement équivalent de points publics ECC distincts.

Nous organisons les findings en trois classes de modes de défaillance (sections 3, 4, 5), précédées de la section 2 qui documente *ce que nous avons cherché et n’avons pas trouvé*. La section négative est méthodologiquement importante : elle borne les revendications des findings positifs et démontre que la chaîne de détection ZetaCert Research est bien calibrée sur le corpus large, et non simplement ajustée pour faire émerger les faiblesses que nous rapportons ensuite.

2. Méthodologie et résultats négatifs

2.1 Détecteurs

Quatre détecteurs ont été exécutés à l'échelle du corpus :

Batch GCD (Bernstein 2004). Sweep PGCD pairwise sur l'ensemble des modules RSA, conçu pour faire émerger les facteurs premiers partagés issus de RNG cassés ou seedés au moment de la génération de clé. C'est le détecteur canonique utilisé par Heninger et al. (2012) pour faire émerger 0,5 % de clés faibles dans la PKI Web de 2012. Exécuté sur le sous-ensemble dédoublé de 6,6 millions de modules de notre corpus ; durée d'exécution ~7 heures sur matériel standard.

Sweep Fermat premiers proches. Détecte les modules RSA générés sous la forme $p \times q$ où $|p - q|$ est suffisamment petit pour que l'algorithme de Fermat retrouve la factorisation en quelques secondes. Cible une classe différente de RNG cassé que batch GCD : lorsque les deux nombres premiers sont tirés du même pool biaisé, ils sont proches en valeur absolue mais non partagés. Exécuté à profondeur $k=100$ sur 25 804 350 modules (10 minutes 38 secondes, ~40 500 modules/seconde sur 12 workers), puis additionnellement à profondeur $k=2\,000\,000$ ($20\,000\times$ plus profonde) sur le sous-corpus le plus à risque — les 30 050 modules issus du scan IoT actif.

Hidden Number Problem / récupération ECDSA par réduction de réseau. Récupération par réduction de réseau d'une clé de signature ECDSA à partir d'une population de signatures avec un nonce biaisé (construction classique Boneh-Venkatesan / Nguyen-Shparlinski). Exécutée deux fois : d'abord sur 23 groupes (émetteur, courbe) constitués depuis les signatures de certificats end-entity visibles via CT (61 tentatives lattice à biais $\in \{2, 4, 8\}$) ; ensuite sur 1 436 signatures extraites en direct depuis des handshakes TLS contre les cinq clusters opérateurs identifiés en section 4.

Scan de mésémission CA. Scan strict-regex pour les certificats émis publiquement dont le Common Name du sujet pointe vers un espace de nommage privé (RFC 1918 IPv4, `.local`, `.internal`, `.lan`, `.home`, `.corp`, `.intranet`, `localhost`). Conçu pour faire émerger les violations de la CA/B Baseline Requirements §7.1.4.2.1.

2.2 Résultats

Détecteur	Population	Hits
Batch GCD	6,6 M modules RSA uniques	0 facteur partagé
Fermat ($k=100$)	25,8 M modules RSA uniques	0 paire de premiers proches
Fermat ($k=2$ M, sous-corpus IoT)	22 000 / 30 050 (interrompu utilisateur)	0 paire de premiers proches
HNP / lattice (sigs extraits CT)	23 groupes (émetteur, courbe), 61 tentatives	0 récupération
HNP / lattice (sigs handshake TLS)	1 436 signatures sur 5 clusters opérateurs	0 récupération
Mésémission CA (regex strict)	44,6 M certificats	2 faux-positifs (artefacts AWS-internes), 0 véritable

Les quatre détecteurs n'ont produit aucune récupération cryptanalytique positive contre la portion CT-visible de l'estate commerciale moderne. Nous interprétons ce résultat comme la preuve que les CA commerciales majeures ont implémenté la génération de clés en HSM grade FIPS, la génération de nonces déterministe-ou-HSM (RFC 6979 ou matériel), et la discipline de construction de templates à un niveau qui résiste aux attaques à l'échelle du corpus public. C'est l'équilibre *au-dessus* de la longue traîne. Les sections suivantes documentent l'équilibre *en dessous*.

2.3 Leçon méthodologique — faux positifs HNP issus d'un sanity check vide

Dans une version initiale du détecteur HNP, une expression de sanity check à la ligne 132 de `research/ecdsa_nonce.py` se lisait `if k_candidate.bit_length() > n.bit_length() - bias + 2: break`. Pour la courbe P-256 (`n.bit_length() = 256`) à `bias = 2`, cette expression s'évalue en `> 256`, ce qui est inatteignable pour tout `k < n` valide. Le check ne brisait donc jamais la boucle, et la routine lattice rapportait chaque candidat « court » comme une clé récupérée. La sortie initiale comprenait des prétentions absurdes de récupération de clés de signature pour des CA connues pour opérer en HSM FIPS-validés (Cloudflare TLS Issuing ECC CA 3, plusieurs intermédiaires Google Trust Services, deux Microsoft Azure ECC TLS CAs, ZeroSSL ECC, cPanel ECC, TrustAsia, Certainly, nazwaSSL). Ces « récupérations » étaient entièrement des artefacts du bug de sanity check décrit ci-dessus : aucune des CA listées ne présentait la moindre faiblesse, chacune est un émetteur correctement opéré et adossé à un HSM, et nous les nommons uniquement pour documenter quelles populations saines le check vide a marquées à tort. Après correction de la borne en `n.bit_length() - bias` (strict) et rejet de `x_candidate ∈ {0, n-1}`, la nouvelle exécution a produit le résultat négatif propre rapporté ci-dessus.

Nous le signalons à titre défensif : au moins un finding « weak-CA HNP » publié antérieurement pourrait avoir utilisé du code lattice template avec un bug latent similaire. Quiconque exécute des détecteurs de type HNP à grande échelle devrait vérifier ses bornes de sanity check contre une population CA connue-propre *avant* de tirer des conclusions d'un hit positif.

3. Mode A — Migration RSA vers la longue traîne IoT/firmware

Le finding « Mining Your Ps and Qs » (Heninger et al., USENIX Security 2012) avait montré que 0,5 % des modules RSA de la PKI Web publique de l'époque pouvaient être factorisés par simple batch GCD, les défaillances étant concentrées dans les appliances réseau et les équipements embarqués qui seedaient leur RNG depuis des sources de faible entropie au premier boot. Quatorze ans plus tard, la classe sous-jacente du bug n'a pas disparu d'Internet — mais elle a migré hors des certificats émis par les CA de la PKI Web publique, vers les valeurs firmware par défaut des fabricants, que les CA publiques ne voient ni ne signent jamais.

Les 47 findings du Mode A se décomposent comme suit :

- **37 findings** à longueur de module RSA-1024 (sous la frontière de retrait NIST SP 800-131A de 2014). L'empreinte fabricant affectée comprend une marque majeure de routeurs grand public (Fabricant A ; 11 findings concentrés dans leurs certificats de gestion par défaut), un fabricant US d'équipement réseau d'entreprise (Fabricant B) dont des équipements en Fin de vie restent en production (3 findings), un fabricant coréen (Fabricant D) et un fabricant chinois (Fabricant E) d'équipements télécom, deux fabricants de CPE industriels (Fabricant F, Fabricant G), un fabricant NAS d'entreprise (Fabricant I), un fabricant de wireless industriel (Fabricant J), une suite logicielle d'automatisation métier d'entreprise (Fabricant K), un fabricant US de plateformes serveurs (Fabricant L ; 4 findings à l'intérieur du contrôleur de plan de gestion — le même fabricant qui produit les certificats à overflow de date du Mode C), et un fabricant d'équipement wireless ISP (Fabricant H). La distribution complète des fabricants se trouve en Annexe A.
- **10 findings** d'exposant public RSA non standard. Le cas saillant est un fabricant allemand d'équipement réseau d'entreprise (Fabricant C) avec 22 certificats couvrant 2014 à 2026 — douze ans de releases firmware — qui utilisent tous `e=17` au lieu du standard `e=65537`. La flotte ECC du même fabricant, en revanche, est pleinement conforme. Il s'agit d'un défaut localisé dans une branche SDK, et non d'un déficit organisationnel de discipline cryptographique ; le fabricant a reconnu le finding et l'a inclus comme remédiation planifiée dans des releases firmware à venir.

Plusieurs findings méritent un commentaire spécifique parce qu'ils éclairent des modes de défaillance qu'un reporting purement statistique masquerait :

Le module 1040 bits. Un fabricant d'équipement wireless ISP (Fabricant H) a produit plusieurs certificats de gestion dont la longueur de module RSA est de 1040 bits — trois bits de plus que 1024, avec les bits de

poids fort du module structurés de manière régulière. Il ne s'agit pas d'une taille non-standard intentionnelle ; c'est la signature d'une routine de génération de clés qui construit p et q à partir d'une sortie RNG concaténée-puis-corrigée et qui échoue à imposer proprement la longueur du module. Des certificats portant cette signature apparaissent dans le corpus jusqu'à janvier 2026, indiquant que le chemin firmware sous-jacent est encore actif sur le terrain.

Le certificat auto-signé usurpant une CA retirée. Un certificat de gestion fabricant pare-feu (Fabricant N) observé pendant le scan a pour sujet `CN=<produit-pare-feu-redacted>`, `O=<fabricant-redacted>` et pour émetteur `CN=Let's Encrypt Authority X3` — une CA intermédiaire retirée en 2021 et qui, même durant sa durée de vie opérationnelle, n'a jamais émis de certificats à validité un an avec `e=3` et un Common Name générique. Il s'agit d'un certificat auto-signé dont la logique de template a incorrectement retenu le DN d'émetteur précédent lorsque l'équipement a régénéré sa propre matière de clé. Le contenu cryptographique du certificat (RSA-2048 avec `e=3`) constitue indépendamment un finding du Mode A ; la construction du template constitue un finding du Mode C (section 5). Le cas illustre que les trois modes de défaillance ne sont pas orthogonaux — le même chemin firmware peut produire des défauts dans plusieurs catégories simultanément.

Réponses obtenues lors de la divulgation (Mode A). Des sept fabricants contactés, un a répondu avec une remédiation firmware planifiée et une reconnaissance explicite que le rapport a déclenché une revue interne de la branche SDK ; un a répondu que le matériel affecté est en Fin de vie et qu'aucune remédiation firmware n'est possible (une issue de divulgation coordonnée structurellement différente mais valide à parts égales) ; cinq n'avaient pas répondu à la date de coupure de la v0.1. La section 6 discute les implications politiques de la réponse Fin de vie.

4. Mode B — Concentration volontaire de clés ECDSA chez les opérateurs

Les findings du Mode A sont reconnaissables comme des variantes d'un phénomène documenté (RNG cassé / keygen défaillant, daté de 2012). Les findings du Mode B ne le sont pas. Nous n'avons pas connaissance d'une mesure systématique précédemment publiée du pattern rapporté ici.

Nous définissons le pattern comme suit. Un *opérateur* — un registrar, un fournisseur de protection de marque, un SaaS de génération de leads, une plateforme de parking de domaines, un fournisseur de pages managées — termine TLS pour une population de domaines client. Dans un déploiement architecturé proprement, chaque domaine terminé reçoit une paire de clés générée indépendamment (c'est l'hypothèse architecturale implicite de l'automatisation ACME). Dans l'anti-pattern de concentration opérateur, l'opérateur génère *une* paire de clés ECDSA et la réutilise comme SubjectPublicKeyInfo sur l'ensemble des certificats qu'il demande pour le compte de tous ses clients.

Nous avons identifié six opérateurs indépendants présentant ce pattern dans le corpus 2026 :

Type d'opérateur	Pays	Nombre de certificats	Clés distinctes
Registrar européen (large, focalisé ccTLD) — Opérateur 1	NL	18 866	1
Fournisseur protection de marque / défense de domaine (Opérateur 2)	UK	4 575	1
SaaS marketing / génération de leads (Opérateur 3)	BR	1 144	2

Type d'opérateur	Pays	Nombre de certificats	Clés distinctes
Hébergeur et parking de domaines (Opérateur 4)	CA	298	3
Fournisseur de parking de domaines (Opérateur 5)	DE	190	1
Hébergeur européen pages managées (Opérateur 6)	FR	32	1
Total		25 105	9

Le cluster brand-protection (Opérateur 2) est le plus structurellement significatif. Une seule clé ECDSA P-256 signe simultanément des certificats pour un ensemble non-affilié de portefeuilles défense-de-marque de clients Fortune 500 — des holdings majeurs de médias et divertissement, une coopérative globale de biens de consommation, un éditeur de jeux vidéo, un anneau de typosquats défensifs autour d'une marque pharmaceutique majeure, et un fabricant d'instruments analytiques. Dans le modèle de menace où la clé de l'opérateur est récupérée (par compromission interne, par accès supply-chain hostile au HSM de l'opérateur, ou par un futur break cryptanalytique), l'attaquant gagne simultanément la capacité d'usurpation TLS contre des milliers de domaines client non-affiliés en une seule opération. Les clients ne sont pas conscients de leur destin cryptographique partagé, parce que le choix architectural est invisible à la couche contractuelle — les conditions de service de l'opérateur décrivent la gestion de clés comme un détail d'implémentation interne.

Nous observons en outre deux sous-cas structurels qui méritent d'être documentés :

Le cluster SCADA mono-organisation. Huit hôtes industriels (Fabricant F ; automates programmables, routeurs de communication modulaires et routeurs superviseurs) au sein d'une grande organisation industrielle partagent une seule clé ECDSA. Contrairement aux cas opérateurs, le déploiement est mono-tenant — la concentration cryptographique est interne à une organisation plutôt que de couvrir des clients non-affiliés. Le cas est inclus comme illustration que le même pattern architectural émerge dans des déploiements non-PKI-opérateurs : une décision d'achats consistant à déployer des certificats industriels identiques issus d'un preset fabricant peut produire la concentration de clés comme effet de bord de l'uniformité opérationnelle.

Le cluster « ghost » d'émission de masse. Pendant la fenêtre d'audit, une clé ECDSA P-256 a été utilisée pour générer 1,65 million de certificats Let's Encrypt sur 1,16 million de sujets uniques en dix jours. Un échantillon aléatoire de sept de ces domaines a été sondé au niveau TLS ; aucun n'a répondu. Les certificats apparaissent émis mais non déployés — une empreinte ACME en masse visible via CT sans surface de service active correspondante. Nous documentons ceci comme une observation méthodologique sur le décalage entre le volume d'émission visible via CT et la réalité service-déployé, plutôt que comme un finding du Mode B dans la même classe de risque que les clusters opérateurs ci-dessus.

Vérification de profondeur cryptanalytique. Pour borner la sévérité du finding de concentration opérateur, nous avons exécuté le détecteur HNP / lattice de biais de nonce ECDSA contre les signatures extraites de handshakes TLS pour chacun des cinq clusters opérateurs (section 2). Le résultat 1 436 signatures, 0 récupération est ici le négatif pertinent : la concentration opérateur n'est pas aggravée par un défaut server-side de génération de nonce qui rendrait le single-point-of-failure architectural récupérable depuis l'observation passive du trafic. Le risque demeure structurel — clé unique, point de défaillance unique, large périmètre d'impact — plutôt qu'une vulnérabilité de récupération passive.

Réponses obtenues lors de la divulgation (Mode B). Les six opérateurs ont été notifiés pendant la fenêtre de drafting v0.1 via le canal de contact sécurité publié par l'opérateur (lorsqu'il en existait un) ou via le canal support général de l'opérateur (en l'absence de contact sécurité publié). Les issues seront reportées en v1.0 après expiration des grâces.

5. Mode C — Défauts de conception sur la durée de vie des certificats

Le troisième mode de défaillance n'est ni un résidu de RNG cassé (Mode A) ni un choix architectural de concentration (Mode B), mais un défaut dans le code de manipulation du temps et de construction de template des routines de génération de certificats firmware.

5.1 Le pattern Y2038 d'expiration codée en dur

Le finding le plus opérationnellement conséquent de cette section est un événement d'expiration synchrone à l'échelle d'une flotte, programmé pour 2038-01-19 03:14:08 UTC — la valeur maximale représentable dans un `time_t` signé 32 bits. À l'échelle du corpus, 1 017 certificats actuellement dans la fenêtre d'audit ont `notAfter` fixé à exactement cette valeur. Sur ces 1 017, 775 (sur environ 721 IPs sondées distinctes) se concentrent sur quatre fabricants maintenus dont les templates de premier-boot émettent `notAfter = INT_MAX` comme valeur unique d'échelle flotte :

Catégorie de fabricant	Équipements distincts (IPs)	Nombre de certificats	Classe de produit représentative
Pare-feu d'entreprise / appliance sécurité PME (Fabricant M)	371	399	Appliances UTM (certificat de gestion par défaut, template CN unique)
Réseau grand public / SOHO (Fabricant A)	319	340	28 modèles de routeurs et modems distincts d'un seul fabricant (deux templates CN d'usine)
Pare-feu d'entreprise (large) — Fabricant N	26	31	Pare-feu nouvelle génération, plan de gestion
Vidéosurveillance réseau (Fabricant O)	5	5	Certificats de gestion de caméras IP
Sous-total flotte fabricant identifiée	≈721	775	

Les 242 certificats restants sont des déploiements résiduels mono-hôte avec des CN à espace de nommage privé divers (`localhost`, préfixes RFC 1918, sujets ad hoc) ; ils partagent le déclencheur Y2038 mais pas la structure d'amplification flotte qui fait des quatre clusters fabricants la préoccupation opérationnelle de cette section.

Le fait structurel est qu'à 03:14:08 UTC le 2038-01-19, le certificat de plan de gestion de chaque équipement affecté expire *simultanément* et *mondialement*. Les opérateurs des équipements affectés trouveront, ce matin-là, leurs interfaces web d'administration affichant des avertissements de certificat expiré ; selon la chaîne d'outillage de gestion, le déploiement de configuration automatisé peut échouer fail-closed, les agents de monitoring peuvent refuser la connexion, et les tunnels SSL-VPN se terminant sur le plan de gestion peuvent cesser de renégocier. Aucun des firmwares fabricants affectés ne contient un chemin de renouvellement gracieux — le certificat est généré une fois en usine ou au premier boot et persiste comme identité du plan de gestion pour la durée de vie de l'équipement.

La cause racine est convergente entre les quatre fabricants : le firmware génère un certificat de gestion auto-signé au premier boot et calcule son `notAfter` comme `INT_MAX` — soit explicitement (comme la valeur maximale représentable en `time_t` 32 bits), soit implicitement (comme le résultat d'une arithmétique sur une valeur 32 bits qui déborde vers `INT_MAX`). Le chemin de code précède la migration générale de l'industrie vers la manipulation 64 bits du temps, et n'a pas été revisité lorsque les releases firmware suivantes ont adopté des plateformes sous-jacentes 64 bits.

5.2 Overflow de date dans le sens inverse

Un finding séparé dans la même classe conceptuelle : deux certificats de gestion provenant d'un fabricant US majeur de plateformes serveurs (Fabricant L) ont des valeurs **notAfter** *antérieures* au 1970-01-01. Ces certificats sont techniquement expirés-à-l'émission dès l'instant de leur génération. La cause racine est l'image miroir de Y2038 : une arithmétique sur un `time_t` qui wrap vers une valeur négative, puis qui est rendue dans le champ `GeneralizedTime` du certificat comme une date pré-epoch. La même famille de produits du fabricant contribue quatre findings au Mode A (certificats de gestion RSA-1024) — les deux findings vivent à l'intérieur du même sous-système de contrôleur de plan de gestion firmware.

5.3 Le certificat auto-signé usurpant une CA retirée

Discuté en section 3 aux côtés du finding Mode A avec lequel il co-existe : le certificat de gestion fabricant pare-feu (Fabricant N) de sujet `CN=<produit-pare-feu-redacted>` et d'émetteur `CN=Let's Encrypt Authority X3`. L'interprétation Mode C est que la routine de génération de certificats du firmware a retenu le champ `DN-émetteur` d'une version précédente du template (lorsque l'équipement servait peut-être un certificat émis Let's Encrypt via ACME) sans le nettoyer lors de la régénération. Le certificat est cryptographiquement auto-signé mais se présente DN-wise comme s'il avait été émis par un intermédiaire commercial défunt. Le défaut est dans le code de construction de template, et non dans les primitives cryptographiques — Mode C, et non Mode A.

5.4 Pourquoi il s'agit d'une seule classe de mode de défaillance

Expiration Y2038 codée en dur, overflow de date pré-1970, et artefact de construction de template paraissent superficiellement non liés. Nous soutenons qu'il s'agit de la même classe parce qu'ils partagent une propriété structurelle : le *contenu* du certificat est faux d'une manière que la *cryptographie* du certificat ne signalerait pas. Chaque défaut est invisible à un check de robustesse de clé, à un check de vérification de signature, à un check de validation de chaîne, ou à un check de suite cipher. Ils ne deviennent visibles que lorsqu'un outil inspecte le contenu du certificat avec une conscience sémantique — que le champ **notAfter** est implausible, qu'un DN d'émetteur nomme une CA défunte, qu'une valeur **notAfter** identique apparaît sur des certificats issus d'équipements non liés.

C'est une catégorie de bug pour laquelle la cryptanalyse mature n'a pas été construite. C'est cependant exactement la catégorie de bug qui devient opérationnellement conséquente à grande échelle — l'expiration synchrone à l'échelle de la flotte Y2038 à elle seule est une feuille de route vers une panne 2038 qui est, aujourd'hui, à douze ans d'inévitabilité et entièrement traitable par mises à jour firmware émises pendant cette fenêtre.

Réponses obtenues lors de la divulgation (Mode C). Le plus grand finding mono-fabricant a été divulgué au PSIRT du fabricant pendant la fenêtre de drafting v0.1 avec une empreinte de 144 IPs au moment de la divulgation ; la croissance continue du corpus a depuis porté la flotte observable à ≈ 371 IPs. La divulgation reste exacte en tant que rapport de pattern de conception — la mise à jour du compteur est un artefact de fenêtre de mesure, pas une nouvelle vulnérabilité — et ne sera pas réémise. L'extension cross-vendor ($\approx 319 + 26 + 5$ IPs provenant de trois fabricants supplémentaires) est en cours de vérification de contact PSIRT par fabricant et de divulgation. Les issues seront reportées en v1.0.

6. Spectre des réponses obtenues lors de la divulgation

Une campagne de divulgation coordonnée à cette échelle produit un spectre de réponses fabricants. Nous documentons les quatre points du spectre représentés dans l'audit 2026, parce qu'ils ont des implications politiques au-delà des findings spécifiques.

Réponse 1 — Reconnaissance et remédiation planifiée. Le fabricant allemand d'équipement réseau d'entreprise (Fabricant C) dont le cluster `e=17` de 22 certificats est documenté en section 3 a répondu en quatre jours, a positionné le finding comme une affaire de « bonne pratique / hygiène numérique » plutôt

que comme une vulnérabilité en propre, a accepté l'attribution-par-empreinte sans remettre en cause la méthodologie, et s'est engagé à ajuster l'exposant dans les releases firmware à venir. Il s'agit de la réponse de divulgation la plus propre possible : le fabricant reçoit, évalue, remédie, et communique le calendrier.

Réponse 2 — Reconnaissance et incapacité structurelle de remédier. Le fabricant US d'équipement réseau d'entreprise (Fabricant B) dont les 3 findings RSA-1024 sont documentés en section 3 a répondu que le matériel affecté est en Fin de vie et qu'aucune mise à jour firmware n'est possible. Ce n'est pas un refus de remédier — c'est une déclaration que la politique de cycle de vie du fabricant empêche structurellement la remédiation, et le finding existe alors dans une classe que la divulgation coordonnée ne peut pas résoudre. L'implication politique est que les défauts de cryptographie faible héritée dans du matériel en Fin de vie nécessitent des chemins de remédiation réglementaires plutôt que pilotés par le fabricant (contrats de support étendu, exigences de cryptographie-sunset) à la Fin de vie inscrites dans les frameworks de conformité comme le Cyber Resilience Act de l'UE et NIS2).

Réponse 3 — Déflexion initiale puis recadrage. Une soumission médiée par une plateforme bug-bounty a produit une réponse initiale « ne satisfait pas le seuil pour revue ultérieure, pas de preuve d'exploitabilité réelle ». La réponse est une réponse-type d'un programme bug-bounty face à des findings de nature compliance (la sévérité d'un finding est jugée par son exploitabilité plutôt que par son impact politique ou de conformité). Recadrer la soumission en divulgation-coordonnée-non-bounty et fournir un cinquième finding corroborant (un module 1040 bits de 2019, illustrant que le pattern de défaut keygen est cross-période) a modifié la classification de la plateforme sur le rapport.

Réponse 4 — En attente. Plusieurs divulgations envoyées pendant la fenêtre d'audit n'ont pas produit de réponse avant la coupure de la v0.1. Cela rentre dans la fenêtre normale de divulgation coordonnée à 90 jours. La v1.0 de ce document reportera les issues finales après expiration de grâce 2026-07-25 sur l'ensemble des batches.

7. Guide de détection pour RSSI et achats

Les findings des sections 3-5 ne sont pas détectables par les scanners de suite cipher TLS standard ni par les scanners de vulnérabilités opérant contre les bannières service des équipements. Ils nécessitent une inspection sémantique du contenu des certificats. La checklist d'audit suivante capture les patterns de détection en pratique :

Pour le Mode A (migration RSA). 1. Pour chaque endpoint de terminaison TLS dans votre périmètre et votre plan de gestion, extrayez le certificat et parsez les paramètres de clé publique. Marquez tout module RSA de longueur autre que 2048, 3072, ou 4096 (les tailles légitimes en 2026 sont ces trois ; toute autre indique soit un legacy pré-2014, soit un défaut keygen). Marquez tout exposant public RSA différent de 65537. Marquez toute longueur de module non divisible par 8 (ex. 1040 bits) comme signature probable de défaut keygen. 2. Recoupez les endpoints marqués avec votre inventaire d'équipements par fabricant et modèle ; les certificats firmware par défaut se concentrent par classe d'équipement, donc un finding marqué isolé est probablement indicatif d'une condition à l'échelle de la flotte pour ce modèle.

Pour le Mode B (concentration opérateur). 1. Pour chaque domaine dont la terminaison TLS est déléguée à un opérateur externe (certificats gérés par registrar, certificats gérés par service de protection de marque, landing pages gérées par SaaS de génération de leads, pages parkées gérées, hébergement de pages managées), récupérez le certificat servi sur chaque domaine et extrayez le SubjectPublicKeyInfo. 2. Si plusieurs domaines délégués servis par le même opérateur partagent un SubjectPublicKeyInfo, l'opérateur présente le pattern Mode B. Engagez le canal de contact sécurité de l'opérateur et demandez confirmation architecturale de la présence de génération de clés par-domaine dans leur roadmap produit.

Pour le Mode C (défauts de conception sur la durée de vie). 1. Pour chaque endpoint de terminaison TLS, extrayez le champ `notAfter` du certificat. Marquez tout `notAfter` égal à 2038-01-19 03:14:07 ou 03:14:08 UTC comme candidat Y2038-codé-en-dur. Marquez tout `notAfter` antérieur au 1970-01-01 comme candidat overflow de date. 2. Pour chaque certificat de gestion auto-signé, inspectez le DN d'émetteur. Marquez tout DN d'émetteur qui nomme un intermédiaire CA publiquement reconnu (un certificat auto-signé

ne doit pas présenter un DN d'émetteur de CA publique ; ceci est par définition un défaut de construction de template ou une mauvaise configuration).

ZetaCert opère un service d'audit gratuit et sans compte à audit.zetacert.com qui exécute les trois classes de checks. Soumettez un nom d'hôte ou uploadez un certificat encodé PEM ; le service retourne la même sortie de détecteur qui a produit les findings rapportés dans ce document.

8. Recommandations

Pour les RSSI et architectes sécurité. 1. Ajoutez l'inspection sémantique du contenu des certificats (et non simplement le scan de suite cipher) à votre cadence d'audit périmétrique. Les trois modes de défaillance de ce document sont invisibles aux scanners de suite cipher. 2. Ajoutez une question contractuelle « architecture de génération de clés » à votre processus d'achat et de gestion de fournisseurs pour tout opérateur tiers terminant TLS pour votre compte. La concentration opérateur du Mode B est invisible à la couche contractuelle à moins d'être explicitement questionnée. 3. Ajoutez une requête d'inventaire pour `notAfter` $\in$ [2038-01-19 03:14:07 UTC, 2038-01-19 03:14:08 UTC] et `notAfter` < 1970-01-01 à votre outillage de gestion de certificats. La falaise 2038 est à douze ans et traitable par remédiation ; l'identification préemptive des équipements affectés dans votre flotte aujourd'hui permet de planifier les mises à jour firmware contre les fenêtres de remédiation fabricant. 4. Pour le matériel en Fin de vie exécutant de la cryptographie faible (Réponse 2 en section 6), la réponse structurelle est le remplacement sur un calendrier aligné sur les dispositions de cryptographie-sunset de votre framework de conformité, et non la remédiation fabricant. Maintenez un calendrier publié-et-tracké de fin-de-vie-cryptographique pour votre inventaire d'appliances.

Pour les fabricants générant des certificats auto-signés en firmware. 1. Auditez le chemin de code de génération de certificats pour toute arithmétique sur le temps utilisant un `time_t` 32 bits. Tout appel à `time()` ou arithmétique sur `time_t` en forme 32 bits est un risque Y2038 indépendamment du support 64 bits de la plateforme sous-jacente. 2. Remplacez les certificats de gestion auto-signés à validité fixe par une validité tournante (régénérée périodiquement sur un calendrier de renouvellement gracieux, avec un chemin de fallback-vers-fresh à expiration). Le modèle émission-unique-en-usine produit des risques d'expiration synchrone à l'échelle de la flotte indépendamment de la date d'expiration choisie. 3. Auditez le code de construction de template pour la rétention de champs DN-émetteur entre régénérations de certificats ; chaque régénération doit construire le DN d'émetteur à partir de l'état actuel de l'équipement, et non depuis des valeurs de génération antérieure mises en cache.

Pour les régulateurs (CRA, NIS2, équivalents). 1. Les frameworks de conformité devraient inclure des dispositions explicites de cryptographie-sunset à la Fin de vie : un fabricant qui livre un équipement avec de la cryptographie faible et qui met ensuite fin au support firmware hérite d'une incapacité structurelle à remédier, que la politique de cycle de vie du fabricant seule ne peut résoudre. La réponse réglementaire est un intervalle sunset défini après lequel des équipements à cryptographie faible ne peuvent plus être opérés en contextes régulés.

À propos de ZetaCert Research

ZetaCert est l'entité umbrella derrière PKIFactor (gestion du cycle de vie des certificats), ZetaCA (autorité de certification post-quantique), et CryptoAudit (service public et gratuit d'audit de clés publiques). ZetaCert Research est l'équipe responsable de la campagne d'audit rapportée dans ce document.

Auditez votre propre flotte de certificats sur audit.zetacert.com. Le service est gratuit, ne requiert pas de compte, et applique la même chaîne de détecteurs reportée en section 2 contre un certificat uploadé ou un nom d'hôte sondé.

Pour les requêtes de divulgation coordonnée : vulnerabilities@zetacert.com. Pour les requêtes commerciales (PKIFactor, ZetaCA) : sales@zetacert.com — traitées séparément de toute correspondance de

divulgateur.

Version 1.0 — publication publique, 2026-10-11. Les identités des fabricants et opérateurs sont anonymisées dans toute cette édition ; chaque partie concernée a été notifiée indépendamment via des canaux de divulgation coordonnée (vulnerabilities@zetacert.com) avant publication. Les champs de sujet de certificat susceptibles d'identifier un fabricant sont affichés redactés pour la même raison.

Changelog v0.1 → v0.2 : chiffres Y2038 du Mode C rafraîchis sur le corpus live. Le compte flotte fabricant identifiée passe de 582 certificats / 515 équipements à 775 certificats / ≈ 721 IPs à mesure que le scanning a étendu l'empreinte observée des quatre templates ; les 242 certificats `INT_MAX` mono-hôte résiduels hors des quatre templates fabricants sont reportés séparément, ce qui porte le total des certificats à validité `INT_MAX` dans le corpus à 1 017.