

My Little Certificate

Synthèse de résolution – Certificate Transparency (CT) / TLS



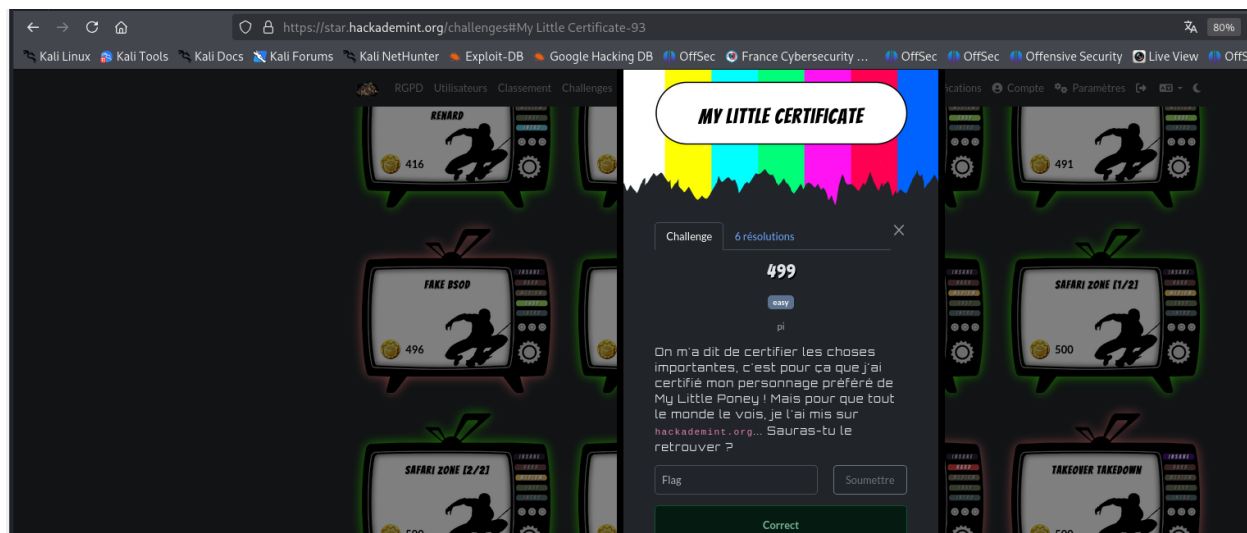
1. Résumé du challenge

Le challenge « My Little Certificate » part d'une histoire volontairement naïve : quelqu'un a été invité à « certifier les choses importantes » et a donc émis un certificat TLS pour son personnage préféré de *My Little Pony*, puis l'a « mis sur hackademint.org ». L'objectif consiste à retrouver ce personnage/indice à partir des journaux publics de Certificate Transparency (CT), puis à le convertir en un flag au format ``Star{...}``.

Indice implicite : le titre « easy pi » renvoie au vocabulaire de la ****PKI**** (Public Key Infrastructure) et au monde des certificats X.509, plutôt qu'à un accès web direct.

Point clé : un sous-domaine peut apparaître dans un certificat (donc dans CT) sans pour autant être réellement déployé en DNS/HTTP. Un retour ``502 Bad Gateway`` ou un domaine non résolu n'invalide pas l'indice : le certificat, lui, existe bien.

Capture – énoncé (référence)



2. Notions utiles

Les éléments suivants suffisent à résoudre le challenge :

- Certificate Transparency (CT) : journaux publics qui enregistrent les certificats émis pour un domaine.
- SAN (Subject Alternative Name) : extension X.509 listant les noms DNS couverts par le certificat.
- CN (Common Name) : nom principal du certificat (souvent redondant avec le SAN).
- Chaîne de confiance : signature de l'AC (ex. Let's Encrypt) et période de validité (notBefore/notAfter).
- My Little Pony (MLP) : franchise animée. Les noms de personnages (ex. Rainbow Dash, Pinkie Pie) servent d'indice lisible.

3. Collecte : interroger les logs CT

Deux approches pratiques (équivalentes) permettent d'obtenir les certificats émis pour un domaine : crt.sh (interface CT) et l'API CertSpotter (SSLMate).

3.1 Recherche via crt.sh

```
# Rechercher tous les certificats contenant des sous-domaines hackademint.org
```

```
# (le caractère % fonctionne comme un wildcard SQL sur crt.sh)
```

```
# À ouvrir dans un navigateur :
```

```
crt.sh/?q=%25.hackademint.org
```

3.2 Recherche via CertSpotter

```
# API CertSpotter (JSON)
```

```
curl -s 'https://api.certspotter.com/v1/issuances?domain=hackademint.org&include_subdomains=true&expand=dns_names' | jq .
```

Dans les deux cas, l'objectif est d'identifier un nom de domaine « parlant » dans la liste des DNS names (SAN), contenant un indice lié à My Little Pony.

4. Filtrer et isoler le certificat pertinent

À partir du JSON, on filtre les entrées dont les SAN contiennent des mots-clés (nom de personnage). Dans vos résultats, un nom ressort : ``star-rainbowdashmybeloved.hackademint.org``.

Exemple de filtrage (CertSpotter + jq)

```
curl -s 'https://api.certspotter.com/v1/issuances?domain=hackademint.org&include_subdomains=true&expand=dns_names'
```

```
| jq -r '[] | select(.dns_names[]? | test("rainbow|pony|pinkie|twilight|flutter|rarity|applejack"; "i")) | .dns_names[]'
```

Captures - extraction du sous-domaine dans le JSON (référence)

```
JSON    Raw Data    Headers
Save Copy Collapse All Expand All Filter JSON

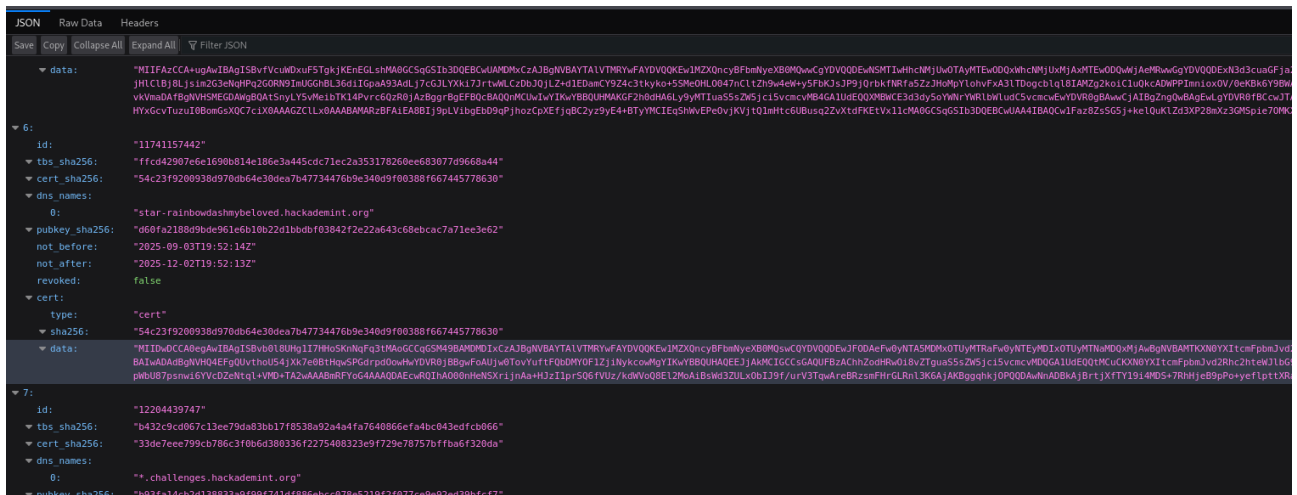
▼ 0:
  id: "11436183187"
  tbs_sha256: "d96bb08e9c95245f6bbcb4b19cf71fe580e2f861ecca8ab24c5cc49acfc5080882"
  cert_sha256: "392abb2cc3clccc6e4cal109d031fa80207c4155f5fb3eb3baa106e6cf23c3"
  dns_names:
    0: "*.challenges.hackademint.org"
  pubkey_sha256: "e05180f1ae0351d8a9ae983857436b2f3ese83be6981577c33a5fc94"
  not_before: "2025-08-09T19:42:41Z"
  not_after: "2025-11-07T19:42:40Z"
  revoked: false

▼ cert:
  type: "cert"
  sha256: "392abb2cc3clccc6e4cal109d031fa80207c4155f5fb3eb3baa106e6cf23c3"
  data: "MIIEFzCCA/upgAdIgaISj53lCnCNySIA3j1Rw3tV3mA8GCSqGISIo3QEBDCADWDMxGzA3BgVVAYTAITVMRywFAVDYQQKEIMXZYowFbmlywKBOMQwwcgYDQVDDQEWSMTEdwhNMjUwODASNTk0MjUxMTA3MTk0MjUwJANHSUwZWYDQVDDDBgBAGAwMEYLvP6pxu1SEgc3PHH16byja4MLw7Gg8BUwA7rvZonZt1FAoAg9CGXBm73UekaycgGegY/2oXLqLgraftbkvGVctsfpsSH5iphSt1FR49LP8e9dnTAAWA8wgqv1qGLILCwlIdQAABGA4TCzCAACdgYVRPAOh/BAAQDAgWgMBGA1UzJQNW80OCcsGAQBFErCjCCCFtArBgnHNSHGedwBTffdaNsTDmkytJitTARBoypBFBDQAPKCBw7TwYBB8BMHAGFZubodAgLybMTTEuSSZmk5ClEvkcWCcAIUEEDgpb6GCKOw3GAcXbnatCj5oMhwYPR1mtUarfcvccscgwYVR0QbAbwcJAIBBgBqhgbAgSRHbwSVdaQAAhBP30FDQK/ZmO/RBJEzy9OT04JOG1YadALty1TOAAAIZONIXAAAEbMGECIFYub5jnRUETQdqeAp5wFOKWpyLAw+ony4rsSiAJAKAoR1gvry13rDPd/3lr1+qJhmldrAfjr/vUT/CFKITANBgkhkIG9wBAQcFAAOCAQEARytYGC19760"

▼ 1:
  id: "11611880154"
  tbs_sha256: "395431402bce8b591fd727855c6ea5013af1do7512a7b01f1ca4d1271987b41"
  cert_sha256: "374a0b4507b14220819858104e96de226a966a422b129a1b7d5592ca3a169"
  dns_names:
    0: "**dev.hackademint.org"
  pubkey_sha256: "28d12ef9bba228f443ae685c6eedabcebfb92b669d942167becb99dd4177e8"
  not_before: "2025-08-24T06:55:47Z"
  not_after: "2025-11-22T06:55:46Z"
```

JSON	Raw Data	Headers
Save	Copy	Collapse All
Expand All	Filter JSON	
▼ data:	<pre>"MlFAzC4Acpag4lBqg1SBvYVcndHx5TgkJKxENGLzhM8GCSqGIsD30DEChWAwQwKcA3JpWbYATLWlVRWYvFYdVQ0KvEJmXZ0nqYfBmlyeXB0WwCpYdVQ00EvWSMTIwncMTJlUw0tAytHwE000vncWUJyAxtHwE00WwJAelMwWqCpYdVQ00EvXJd3cuaGfJyJHlCtJlBjLjsim2G3nHqPzG0R0N1uMUGhL36d1Gpa9A3AdLj7CgJLYXk17jrtwCjD0bJ0JLz+clEDnAwC9Z4c3kyko+S5MEHlD047nclt2n9u4wEvYsFbK3jP9j0tkfNFRfA5Z2JhMqMj1oHvFkA3LTDogbcLq181AMZg2ko1C1u0kADWPP1nn1ox0V/00K8bEY9BwYkVmbAdBfgHqVH5MEGAW0B0AtS1YMe1gBtkd14PvrC6QZ0R8jBqFEBQcBAQ0MhUwJvYkVwBB00HMAKGf2bHdAaHl9yMTlUa5S5Zw5jclSvncvYMB4A1UdE0Q0XMBKcS3d3y5oWmYrYR1lbnUdSc5vncvYdVVR0gBAwCJATBgzngQwBAGEvLgYdVVR0RfBCvcJ7tHYGcTuztU108mGcsX0C7c1X0AAABAGZCLX0AAABABRcfBALEABTBj9pLvlbgEd9pQJh0zcxPEfjgBC2yzyE4+BTYtYtEQSHwMEpEv0KJv101Mhtc6UBusqZ2XvctFKKEVx111CM4GCSqGSI30DEChWAAIA1BA0Cw1Faz8Z5G5j+ke10UK1Zd3XP2Bw3ZGSp5e170MK</pre>	
▼ 6:		
id:	"117411557442"	
tls_sha256:	"ffcd29076e1690b814e186e3a445cdc71ec2a353178260ee683077d9668a44"	
cert_sha256:	"54c23f920093bd97db64c30dea7b47734476b9e340d9f00388f667445778630"	
dns_names:	0:	
	"star-rainbowdashmybeloved.hackandint.org"	
pubkey_sha256:	"d60fa2188d9bde91e610b22d1bbdbf03842f2e22a643c68ebcac7a71ee3e62"	
not_before:	"2025-09-03T19:52:14Z"	
not_after:	"2025-12-02T19:52:13Z"	
revoked:	false	
cert:	"cert"	
sha256:	"54c23f920093bd97db64c30dea7b47734476b9e340d9f00388f667445778630"	
▼ data:	<pre>"MlID0CABAgBgkqhkiG9w0BBQg1I7Hw0S0hRfGf31Hw0CQcGSM9B9ABW0L1CzA3JpWbYATLWlVRWYvFYdVQ0KvEJmXZ0nqYfBmlyeXB0WwCpYdVQ00EvYJF0dAePwYvNTASND0TUyWtRfA5y0tEYyYdIxoTUyWtHd0vWwJAePgnWbMTK8M0Y1tccFpmJlBvBAT0wBAdBfgHqVH5MEGAW0B0AtS1YMe1gBtkd14PvrC6QZ0R8jBqFEBQcBAQ0MhUwJvYkVwBB00HMAKGf2bHdAaHl9yMTlUa5S5Zw5jclSvncvYMB4A1UdE0Q0XMBKcS3d3y5oWmYrYR1lbnUdSc5vncvYdVVR0gBAwCJATBgzngQwBAGEvLgYdVVR0RfBCvcJ7tHYGcTuztU108mGcsX0C7c1X0AAABAGZCLX0AAABABRcfBALEABTBj9pLvlbgEd9pQJh0zcxPEfjgBC2yzyE4+BTYtYtEQSHwMEpEv0KJv101Mhtc6UBusqZ2XvctFKKEVx111CM4GCSqGSI30DEChWAAIA1BA0Cw1Faz8Z5G5j+ke10UK1Zd3XP2Bw3ZGSp5e170MK</pre>	
▼ 7:		
id:	"12204439747"	
tls_sha256:	"b432c9cd067c13ee79da83bb17f853ba92a4af7a640866efa4bc403edfcb066"	
cert_sha256:	"33de7ee99cb786c3f0b6d380336/2275408232f729e78757bf7ba0f320da"	
dns_names:	0:	
	"*.challenges.hackandint.org"	
pubkey_sha256:	"b2361c6b4318823a060d7c1198f8e9ae070cf31042077c0e03e039c7cf7"	

JSON	Raw Data	Headers
Save	Copy	Collapse All
Expand All	Filter JSON	
revoked:	false	
cert:		
type:	"cert"	
sha256:	"374a0b4507b142208198958104e964d6296a966a42b22169a0f07d5592ac3a4169"	
data:	"MIIEFTCCA=Zg9wIBAgISB17063Uf3yynddgV24dJLMMAGCSGqSb3E0QBCwJAMQhCzA3Bg9wBAYTALTRVRYeAYDVQDQeJlMZX00wCFBmlyeyXB00wCgYDVQDQeJlNHTHwMh4wMjI0MDY1OTQ3OzBhWJlUHTlYMDY1OTQ2MzAgR4YDVQDQ0BQlRlLd15oZDp0dG9kRyR5aSAhZjE17C0mXoQ0Mc0M0NTkKc0QwLWbna0H1TRVa3QXp+Py6r17Pm1aSu0YkV503+SRGh17v+0j3b0Yk+vY1e4DR+29nZRNf+J09rRB0861TDTAFNB8SLXE9aUkN3bettCoZpMFK+JuPLpdr+Hh03DK0YtKE46+1eBw/5r5Xl+beVj+X8Pw0CMAH4QYDVROBBYEFK0H+ftntu0Q0cCZnU/C35o4K1IMBGA1Uu1uYQMBAAQForm8bM68T015PeH1yA471pIzMDGCCGSAQUFBwEBBCcw3TAjB9rBGF80Qcwa0YXhR0RdovL3I1wY5pLxmbNyLn9yZy8wIAYDRRB8Bkw41VK15KZyuagfJ2fKw1pb0ub3z1pf057AwaXqB0kMYTqsn5XA1gWkc30SNXaM5FOU2RmCARFCXK0k0pdcLnsM03EAdga4S8U4b0wVNY1T0GGc0PP3+T+0eYoeIn6B0YTR+SYm0GAZzjDE1eU0AAEAwBHEUCIT000wRAF5yef4bYsa0skjfe0mlrSK3mN9yZ1r0z3c1K0j1J10n5F8MwR3hV3ZmVMM44RD44k8A="	
2:		
id:	"11611886076"	
tbs_sha256:	"c09f2380258f2e06b74cea3359c10916ba3662df2d25be2a0d36f8142fd46f9"	
cert_sha256:	"29f6a054c1fbee55546fc3d66261268c8f7de020e32ae6377b009f79402b"	
dns_names:		
0:	"hackademint.org"	
pubkey_sha256:	"ac2d08ce3118de3e5133c82a537d49e48c88043bd1fbafe949b7eba6bb8fd"	
not_before:	"2025-08-24T06:56:25Z"	
not_after:	"2025-11-22T06:56:24Z"	
revoked:	false	
cert:		
type:	"cert"	
sha256:	"29f6a054c1fbee55546fc3d66261268c8f7de020e32ae6377b009f79402b"	
data:	"MIIEccCAQ9g9IBAgISB18dbnhtFACBL3b03662+MMAGCSGqSb3E0QBCwJAMQhCzA3Bg9wBAYTALTRVRYeAYDVQDQeJlMZX00wCFBmlyeyXB00wCgYDVQDQeJlNHTHwMh4wMjI0MDY1OTQ3OzBhWJlUHTlYMDY1OTQ3MzAgR4YDVQDQ0BQlRlLd15oZDp0dG9kRyR5aSAhZjE17C0mXoQ0Mc0M0NTkKc0QwLWbna0H1TRVa3QXp+Py6r17Pm1aSu0YkV503+SRGh17v+0j3b0Yk+vY1e4DR+29nZRNf+J09rRB0861TDTAFNB8SLXE9aUkN3bettCoZpMFK+JuPLpdr+Hh03DK0YtKE46+1eBw/5r5Xl+beVj+X8Pw0CMAH4QYDVROBBYEFK0H+ftntu0Q0cCZnU/C35o4K1IMBGA1Uu1uYQMBAAQForm8bM68T015PeH1yA471pIzMDGCCGSAQUFBwEBBCcw3TAjB9rBGF80Qcwa0YXhR0RdovL3I1wY5pLxmbNyLn9yZy8wIAYDRRB8Bkw41VK15KZyuagfJ2fKw1pb0ub3z1pf057AwaXqB0kMYTqsn5XA1gWkc30SNXaM5FOU2RmCARFCXK0k0pdcLnsM03EAdga4S8U4b0wVNY1T0GGc0PP3+T+0eYoeIn6B0YTR+SYm0GAZzjDE1eU0AAEAwBHEUCIT000wRAF5yef4bYsa0skjfe0mlrSK3mN9yZ1r0z3c1K0j1J10n5F8MwR3hV3ZmVMM44RD44k8A="	
3:		



5. Prouver que le certificat est valide (même si le site ne répond pas)

Un certificat présent dans CT atteste qu'une AC l'a émis et qu'il est publiquement journalisé. La « validité » se vérifie ensuite via (1) la période `notBefore/notAfter`, (2) l'absence de révocation, et (3) une chaîne de confiance cohérente vers l'AC.

Vérification locale de CN/SAN (à partir de `cert.data` en base64) :

1) Décoder en DER puis afficher CN/SAN

```
base64 -d cert.b64 > cert.der
```

```
openssl x509 -inform der -in cert.der -noout -subject -issuer -dates -ext subjectAltName
```

Optionnel : vérifier la chaîne (si vous récupérez l'intermédiaire/CA) :

Exemple (selon le bundle AC utilisé)

```
# openssl verify -CAfile chain.pem cert.pem
```

Vous devez retrouver dans `subjectAltName` (SAN) le nom DNS complet `star-rainbowdashmybeloved.hackademint.org`.

6. Déduction du flag

Le sous-domaine contient un message lisible. L'interprétation la plus cohérente avec l'énoncé est de récupérer le ****nom du personnage**** : ***Rainbow Dash***.

Normalisation recommandée :

- Partir du FQDN trouvé dans le SAN.
- Retirer le suffixe `.hackademint.org`.
- Retirer le préfixe technique éventuel (`star-`).
- Extraire le nom du personnage (ici `rainbowdash`) puis adapter la casse si nécessaire.

Candidats typiques à tester (du plus « sémantique » au plus « littéral ») :

- Star{rainbowdash}
- Star{RainbowDash}
- Star{rainbowdashmybeloved}
- Star{star-rainbowdashmybeloved}

Si la plateforme refuse un candidat, cela signifie uniquement que la normalisation attendue diffère (casse, présence du préfixe, ou extraction stricte du nom).

7. Pièges et erreurs fréquentes

- Confondre indisponibilité HTTP (502) et absence du certificat : CT prouve l'émission, indépendamment du service web.
- Répondre avec une empreinte SHA-256 (cert/pubkey) alors que l'énoncé attend une chaîne lisible.
- Inclure le wildcard `*.` ou le domaine complet alors que seule l'étiquette utile est attendue.
- Oublier de tester la casse (RainbowDash vs rainbowdash).

Annexe A – Extrait indiciel

Dans nos résultats CT, l'entrée qui contient l'indice mentionne explicitement : `dns_names: ["star-rainbowdashmybeloved.hackademint.org"]`.



Certificate Transparency (CT)

CT est un mécanisme de “journalisation publique” des certificats TLS/SSL : lorsqu’une autorité émet un certificat, celui-ci est généralement **enregistré dans des logs CT append-only** (gérés par divers opérateurs).

Objectifs principaux :

- **Détecter** les certificats émis par erreur ou de manière frauduleuse (ex. pour un domaine qui ne vous appartient pas).
- **Rendre auditable** l'écosystème des certificats : on peut rechercher quels certificats ont été émis pour un domaine, même si le site n'est pas accessible.

En pratique, CT permet donc de retrouver des **noms de sous-domaines** présents dans le certificat (souvent via le champ SAN), même si ces sous-domaines ne répondent pas en DNS/HTTP.

AC (Autorité de Certification)

Une AC (en anglais CA, Certificate Authority) est l'organisme qui **émet et signe** des certificats numériques X.509 (TLS). Elle :

- vérifie (à un certain niveau) que le demandeur **contrôle le domaine** concerné (validation DV/OV/EV selon le type),
- signe le certificat avec sa clé privée, ce qui permet aux navigateurs/systèmes de **faire confiance** au certificat via une chaîne de certification (AC intermédiaire → AC racine).

En résumé : l'**AC émet/sign**e, et **CT publie/consigne** pour audit.