

My Little Certificate

Synthèse de résolution – Certificate Transparency (CT) / TLS



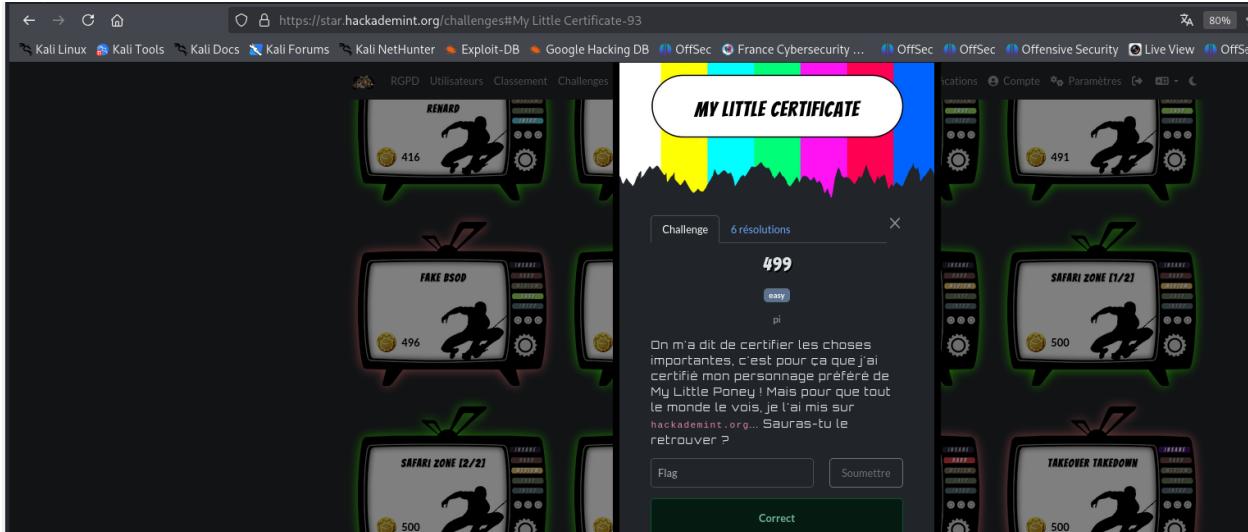
1. Résumé du challenge

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
<pre>Save Copy Collapse All Expand All ▾ Filter JSON</pre>	<pre>0: id: "11436183187" tbs_sha256: "d968b008c95254f6bbcb4b19cf19e580e2fb81ecc8ab24c5cc49acf5080882" cert_sha256: "392dbb23c3cc6e4ca109d03faf80207c4155f5fb3e63b3aa106e6fc23c3" dns_names: 0: "**.challenges.hackademint.org" pubkey_sha256: "e051580fe1ace0353c8b89ae9983857436b2f3eae83b6e98157e773c3a5fc94" not_before: "2025-08-09T19:42:41Z" not_after: "2025-11-07T19:42:40Z" revoked: false cert: type: "cert" sha256: "392dbb23c3cc6e4ca109d03faf80207c4155f5fb3e63b3aa106e6fc23c3" data: "MIIFErzCA/cupAHgkIBA0gBz581CnCrnCYSAIj3aWQGKzAjBgNVAATLVTRMyFAYDV00Kew1MZX0ncyBFmNyxB0M0wwCgJYDVO00DeNSMTIwHhCNMjUwOD1ASMTk0Mj0xWhhNMjUxMTA3MTk0Mj0wMjAaMSlwTwYDV000BwqLmtN0YwxsZwBeL1VpSpvX15EgCFHH16byasLMV7g0B0uAt7rvZon2ntjFa0dpGcXBMy3lUusAsyqGxg9y2XoUelgrafbkqg1Vctsrf55Hs4pnts1lRu49LP8DdknIAVAbwvq1GLvt1lcszIDAOABe1Ck2z1Ca1cogDwRPA0H/BAD0AgpMBGAl1uJ0QwB0GCCgA0UfFrcfCIICATB0nVHSHEGDAwgbTFa0x6vTdMp510t4fSLsvb1tAzbppgBgEF0cBA0D0MC1wvTkwBB00UMAKGf2h0dA6l9yMTeu55szWSj15vcmcvMB4GA1UdE00pMBEChCovYzhbGx1bm1c5yvMnR1bW1ld5vcncvEwYDwR0BawcJA1BqZng0wBaqFFsRRhobv5Ad0A0BP90F0dQ/2oM0/8djEZy90704j0GiyaxdALtyZ0AAAZ1k0KwvLa+wNy4rSsiA1Kaorlgy19jT0Pd/31r1+QjhnnidrFjrvT7/CfIKTANBgkqhigSw0BA0sFAAOCAEY3yGKI760</pre>	<pre>Save Copy Collapse All Expand All ▾ Filter JSON</pre>
<pre>1: id: "11611800154" tbs_sha256: "3954314626cc8b591f7d27855c6ea501afda751a7b0f1ca4d1271987b41" cert_sha256: "374a0b4507b142208198950104e96d296a966a42b22169af87d5592ac34169" dns_names: 0: "**.dev.hackademint.org" pubkey_sha256: "28d12e1fbb228443ae685c66edabce0fb92b669d9421678becb99dd4177e8" not_before: "2025-08-24T06:55:47Z" not_after: "2025-11-22T06:55:46Z" revoked: false cert: type: "cert" sha256: "3954314626cc8b591f7d27855c6ea501afda751a7b0f1ca4d1271987b41"</pre>	<pre>Save Copy Collapse All Expand All ▾ Filter JSON</pre>	
<pre>2: id: "11741157442" tbs_sha256: "ffcd429076e1690b814e1863a445cdc71ec2a53178260ee683077d9668a44" cert_sha256: "54c23f00938d970dbd430dea7b7734476b09e340d9f00388f667445776830" dns_names: 0: "star-rainbowdashmybeloved.hackademint.org" pubkey_sha256: "d6f0fa21889b9de961e6b10b22d1bbdbf03842f2e22a643c68ebca7a71ee3e02" not_before: "2025-09-03T19:52:14Z" not_after: "2025-12-02T19:52:13Z" revoked: false cert: type: "cert" sha256: "54c23f00938d970dbd430dea7b7734476b09e340d9f00388f667445776830" data: "MIIE0dCAQeeAwIBAgIBBwIB8UHg117Hb0SkNqFq3tMoCcGqS9M8BMdMD1cxAjBgNVAATLVTRMyFAYDV00Kew1MZX0ncyBFmNyxB0M0wwCgJYDVO00DeNSMTIwHhCNMjUwOD1ASMTk0Mj0xWhhNMjUxMTA3MTk0Mj0wMjAaMSlwTwYDV000BwqLmtN0YwxsZjHClBj8Lj2im23e9npQp026RN91mJGghBL36d11GpaA93Adl7c6JLYXk173rtwLc2d0jLz+d1eDam9Y924c3tjkyo+5SM0hL0847nC1tLz9w4eM+y5FbKj5Pj9jOrbkRfRta5Z23HoMpylonvFxKA3LTDogcb1q18AM2g2ko1Cu0kCAdWPPIml0xv/0exBk6Y9BwvkvMaDwBqVNHSEGDAwgbTFa0x6vTdMp510t4fSLsvb1tAzbppgBgEF0cBA0D0MC1wvTkwBB00UMAKGf2h0dA6l9yMTeu55szWSj15vcmcvMB4GA1UdE00pMBEChCovYzhbGx1bm1c5yvMnR1bW1ld5vcncvEwYDwR0BawcJA1BqZng0wBaqFFsRRhobv5Ad0A0BP90F0dQ/2oM0/8djEZy90704j0GiyaxdALtyZ0AAAZ1k0KwvLa+wNy4rSsiA1Kaorlgy19jT0Pd/31r1+QjhnnidrFjrvT7/CfIKTANBgkqhigSw0BA0sFAAOCAEY3yGKI760</pre>	<pre>Save Copy Collapse All Expand All ▾ Filter JSON</pre>	
<pre>3: id: "12204439747" tbs_sha256: "b432c9b6d6713ee9d08bb17f8538a924a4f7e48408666ca4bc043edfc066" cert_sha256: "33de/eeef799c7b5c3fb0b358033612275408323e91729e78757bf1bae320da" dns_names: 0: "**.challenges.hackademint.org" pubkey_sha256: "b03fa174c2138833a0f99f741df8f86ebcr78e5210f2d077ce9e97e213b9fcf7"</pre>	<pre>Save Copy Collapse All Expand All ▾ Filter JSON</pre>	
<pre>4: id: "11611866076" tbs_sha256: "c0f973000258f2e6b74ce335b9c10916ba3662fd25be2a0d36ff8142df649" cert_sha256: "29f6a0451cfbecc55546f63d66261268c8f7de020e32e6377bd09f79402b" dns_names: 0: "hackademint.org" pubkey_sha256: "ac2d08c3118de31533c82a537d4d9e48c88043bd1fbfae949b7ebaf6bb8fd" not_before: "2025-08-24T06:56:25Z" not_after: "2025-11-22T06:56:24Z" revoked: false cert: type: "cert" sha256: "29f6a0451cfbecc55546f63d66261268c8f7de020e32e6377bd09f79402b" data: "MIIE+zCCA/0gAwIBAgISBa0nbKtFAC0L8aBq180G62gMA0cCSg1b300EBwUAMDbMxCzAjBgNVAATLVTRMyFAYDV00Kew1MZX0ncyBFmNyxB0M0wwCgJYDVO00DeNSMTIwHhCNMjUwOD1B0Y1j1lWhhNMjUxMTIyMDY1j1i0wMjAaMSlwTwYDV000BwqLmtN0YwxsZKRe76lHkL50Y0anhh1AgfBAGggIgHIIHC0A0BgVWH088AEBANCBAwH0yD0R1BbYFAY1kWBB00AwEGCsGa0UfBwMKMwA1UdEwB/0CMhAwh0YDwR0B0BEYFCFL77x0k0CkR1064k4T1l1u5MB6A1U0iU0Y0hBafAC1Kf1tj18x631MrXg+tzp0NHSMDGCGsCAQ0fUbEBBCwTAjBgrBgFB0cvAoTxR0R0dov31xM15pLmx1bNm1yLn1y2BwGyDwR0RBPMwEY1PoGf1azF2x1lpbnQu33wM0MGAl1dIA0mAowcAYGz4EM0Q1BMc4G1A1UhQnMCUw1Gah0dGMh0dAGLyhTi1y5z2Nsj15vcncvEwYDwR0BawcJA1BqZng0wBaqFFsRRhobv5Ad0A0BP90F0dQ/2oM0/8djEZy90704j0GiyaxdALtyZ0AAAZ1k0KwvLa+wNy4rSsiA1Kaorlgy19jT0Pd/31r1+QjhnnidrFjrvT7/CfIKTANBgkqhigSw0BA0sFAAOCAEY3yGKI760</pre>	<pre>Save Copy Collapse All Expand All ▾ Filter JSON</pre>	
<pre>5: id: "11611866076" tbs_sha256: "c0f973000258f2e6b74ce335b9c10916ba3662fd25be2a0d36ff8142df649" cert_sha256: "29f6a0451cfbecc55546f63d66261268c8f7de020e32e6377bd09f79402b" dns_names: 0: "hackademint.org" pubkey_sha256: "ac2d08c3118de31533c82a537d4d9e48c88043bd1fbfae949b7ebaf6bb8fd" not_before: "2025-08-24T06:56:25Z" not_after: "2025-11-22T06:56:24Z" revoked: false cert: type: "cert" sha256: "29f6a0451cfbecc55546f63d66261268c8f7de020e32e6377bd09f79402b" data: "MIIE+zCCA/0gAwIBAgISBa0nbKtFAC0L8aBq180G62gMA0cCSg1b300EBwUAMDbMxCzAjBgNVAATLVTRMyFAYDV00Kew1MZX0ncyBFmNyxB0M0wwCgJYDVO00DeNSMTIwHhCNMjUwOD1B0Y1j1lWhhNMjUxMTIyMDY1j1i0wMjAaMSlwTwYDV000BwqLmtN0YwxsZKRe76lHkL50Y0anhh1AgfBAGggIgHIIHC0A0BgVWH088AEBANCBAwH0yD0R1BbYFAY1kWBB00AwEGCsGa0UfBwMKMwA1UdEwB/0CMhAwh0YDwR0B0BEYFCFL77x0k0CkR1064k4T1l1u5MB6A1U0iU0Y0hBafAC1Kf1tj18x631MrXg+tzp0NHSMDGCGsCAQ0fUbEBBCwTAjBgrBgFB0cvAoTxR0R0dov31xM15pLmx1bNm1yLn1y2BwGyDwR0RBPMwEY1PoGf1azF2x1lpbnQu33wM0MGAl1dIA0mAowcAYGz4EM0Q1BMc4G1A1UhQnMCUw1Gah0dGMh0dAGLyhTi1y5z2Nsj15vcncvEwYDwR0BawcJA1BqZng0wBaqFFsRRhobv5Ad0A0BP90F0dQ/2oM0/8djEZy90704j0GiyaxdALtyZ0AAAZ1k0KwvLa+wNy4rSsiA1Kaorlgy19jT0Pd/31r1+QjhnnidrFjrvT7/CfIKTANBgkqhigSw0BA0sFAAOCAEY3yGKI760</pre>	<pre>Save Copy Collapse All Expand All ▾ Filter JSON</pre>	

```

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

  ▼ data:
    "MIIFAzCCA+ugAwIBAgISBvVcdWxuFStgkKEnEGLshMA0GCSqGSIb3DQEBCwUAMDCzAxBgNVBAYTA1VTMRYwFAYDVQKExIzX0nycBFbmNyeXB0QwvCgYDVQDEwNSMTIwHcNMjUwOTAyMTEwODQwJAcMRowGgYDVQDExN3d3cuGFja
    jH1CBljBLsi+2c3eNqPqZGRN9ImUGhL3d1Gqa93Adt7cGjLYXk1737rWLzCzb30jLz+d1E0mCY924c3tkyko+SMoOHLo47nC1tZh9w4eW+y5FbC9s3Pbj0rbkNRfz5ZJHMpY1ohvFx31TDogcb1q18IAMzg2koiclu0kqADWP1mn1ox0v/0vEBk6y9Bw
    vkvmao4TBgnVMSMGEAgwBA7nly5Me1bK1APvrc602r0A2zBgrBgEPB0cBAQ0McUwT1KwYB0UHMKGF2hdHAGLy9MT1u055zW5c15vcmcMB4GAIude00MBWC3d3dySoYmrWrbw1uc5vcnwEwDV0R6BawwCj1B1BZng0BaGeLgYDRFBcvJT
    HYXGcTuuz10BomGsXQC7c1X0AAAG2CLlxAAABAMRzBFA1EA881j9pLVibgeBD9qPhozcPXEfjqC2y2z9yE4+B7yMCIEq5HvEp0vKV)01m#t6UBusqZ2XtFdKEtVx11cMA0gCSqGSIb3DQEBCwUA4IBAQcWiaz8zSGS)+keQuK1zd3XP28nX23GM5p1e70MK

  ▼ 6:
    id: "11741157442"
    tbs_sha256: "ffcd4290766c16900814e186e3a445cdc71e:2a353178260ee683077d9668a44"
    cert_sha256: "54c23f19200938d970db64e30dea147734476b9e340d9f00388f667445778630"
    dns_names:
      0: "star-rainbowdashmybeloved.hackademint.org"
    pubkey_sha256: "d60f0ea2188d9de9616eb10b2d1bbdbf03842f2e22a643:68ebcac7a71ee3e62"
    not_before: "2025-09-03T19:52:14Z"
    not_after: "2025-12-02T19:52:13Z"
    revoked: false
    cert:
      type: "cert"
      sha256: "54c23f19200938d970db64e30dea147734476b9e340d9f00388f667445778630"
    data:
      "MIIDwDCA66gAwIBAgISBvVcdWxuFStgk177HfHosKnqfj31Ma0GCSqGSIb3DQEwDwIwIKAjBgNVBAYTA1VTMRYwFAYDVQKEv1KZX0nycBFbmNyeXB0QwvC0YDWQDEvJF00AfFw6yNTA5MDIxOTUwMTRaFwOyNTEwDjx0TUwMTNaMDQwMjAwBgNVBAMTN0YX1tcmFbmJvd
      BA1wD4aBgnVMSMGEAgwBA7nly5Me1bK1APvrc602r0A2zBgrBgEPB0cBAQ0McUwT1KwYB0UHMKGF2hdHAGLy9MT1u055zW5c15vcmcMB4GAIude00MBWC3d3dySoYmrWrbw1uc5vcnwEwDV0R6BawwCj1B1BZng0BaGeLgYDRFBcvJT
      phbU07pnw16TVc0zeNtql+VMd+TA2wAAABIRFyGAAAA0DAeCr01hAaHJz1lprS06FvUz+kdWv008El2MoA1Bswd3ZULx0b139f/urV3tqwAreFGrLn13K6A1AKBggkhhJ0PQ0DAwNhADBkA1Br+tXFTY1914MD5+7RHjeB9p+yeLpttXR

  ▼ 7:
    id: "1202439747"
    tbs_sha256: "b432c9c067c13ee79d83bb17f8538a92a44fa7640866ef4dbc043edfcbb66"
    cert_sha256: "33de7eee799c6b786c3f0b6d380336f2275408323e9f729e78757bffb6f1320da"
    dns_names:
      0: ".challenges.hackademint.org"
    pubkey_sha256: "h03fa14ch2d138033a9f90f7d1df86fchry078e5219f2f077cc9e92e139hf-f7"


```

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 auditabile** 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/signe**, et l'**CT publie/consigne** pour audit.