Instructions relatives aux expressions régulières et considérations de performances pour le filtrage des URL

Table des matières

Introduction

Ce document décrit les directives et les considérations de performances pour l'utilisation d'expressions régulières dans le filtrage d'URL avec le moteur UTD. Le filtrage des URL dans le moteur UTD utilise la bibliothèque d'expressions régulières PCRE2.

Contribution d'Eugene Khabarov, ingénieur Cisco.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Syntaxe des expressions régulières (regex)
- Concepts de filtrage URL
- Configuration de Unified Threat Defense (UTD)
- Différences entre les protocoles HTTPS/HTTP

Composants utilisés

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations générales

Bien que PCRE2 soit puissant, certaines expressions complexes ou « gourmandes » peuvent provoquer un retour en arrière excessif et atteindre des limites internes dans le moteur regex. Dans ce cas, le traitement d'un modèle peut prendre trop de temps et être traité comme « aucune correspondance ».

Points clés

- PCRE2 impose des limites internes sur les étapes de retour arrière ou le temps de correspondance afin de protéger les ressources système.
- Certains modèles sont syntaxiquement valides, mais ne sont pas fiables sur le plan informatique et peuvent déclencher un « retour en arrière catastrophique ».
- Lorsque ces limites sont dépassées, le moteur regex peut interrompre le traitement et ne renvoyer aucune correspondance, même si l'URL correspond logiquement au modèle.

Modèles à éviter

Évitez les constructions regex qui combinent :

- Quantificateurs imbriqués, par exemple : (...+)*, (.*)*, (.+)+, etc
- Les caractères génériques (.) se répètent sur de grandes portions de la chaîne, en particulier à la fin du motif
- Points non échappés dans les noms de domaine lorsqu'ils sont utilisés avec répétition

Par exemple, ici, le modèle est syntaxiquement valide mais peut être coûteux à traiter :

```
^([a-zA-Z0-9-]+.)*portal.example.com$
```

Remarque : Dans ce cas, ([a-zA-Z0-9-]+.)* est un groupe avec un quantificateur imbriqué (+ inside *) plus un caractère générique (.). Sur certaines entrées non concordantes, le moteur regex peut explorer un très grand nombre de chemins de retour arrière.

Meilleures pratiques recommandées

Toujours échapper aux points dans les noms d'hôte

Utilisez \. afin de faire correspondre un point littéral, par exemple :

```
^([a-zA-Z0-9-]+\.)*portal\.example\.com$
```

Motifs d'ancrage et caractères de restriction

Utilisez ^ et $ et limitez-vous aux caractères attendus (par exemple, [a-zA-Z0-9-] pour les étiquettes d'hôtes) afin de réduire le retour arrière.

Éviter Les Répétitions Imbriquées Et Illimitées Lorsque C'Est Possible

Préférez les constructions plus simples plutôt que les motifs complexes qui tentent de couvrir tout dans un regex. Considérez plusieurs entrées spécifiques au lieu d'une expression très large.

Modèles de test dans un testeur compatible PCRE2

Avant le déploiement, testez les modèles regex dans un environnement compatible PCRE2 et évitez les modèles qui génèrent des « retours en arrière catastrophiques » ou des avertissements similaires.

Remarque : Si un modèle d'expression régulière atteint les limites internes du moteur PCRE2, il peut être traité comme « aucune correspondance » par le moteur de filtrage d'URL. Dans de tels cas, la classification des URL revient à la catégorie ou à la réputation, et non au résultat de l'expression régulière liste blanche/liste noire. Les limites exactes sont spécifiques à l'implémentation et peuvent varier d'une version à l'autre. Vous devez concevoir les index avec prudence.

Différences de correspondance d'URL pour HTTP et HTTPS

Le moteur UTD inspecte les URL différemment pour le trafic HTTPS et HTTP. Cela affecte la manière dont les expressions régulières doivent être conçues pour le filtrage d'URL.

Trafic HTTPS (TLS)

Pour le trafic HTTPS chiffré, le moteur UTD ne déchiffre pas la charge utile par défaut.

- Le filtrage des URL utilise l'indication de nom de serveur (SNI) du client TLS (Transport Layer Security)Hello.
- Le modèle d'expression régulière est appliqué au nom d'hôte SNI uniquement, par exemple : api.example.com

Dans ce cas, un modèle basé sur le nom d'hôte est mis en correspondance avec la chaîne de nom d'hôte api.example.com telle que :

```
^([a-zA-Z0-9-]+\.)*example\.com$
```

Trafic HTTP (non chiffré)

Pour le trafic HTTP simple, le moteur UTD peut voir la requête HTTP complète (ligne de requête et en-têtes).

Selon l'implémentation, la chaîne donnée au moteur regex peut inclure :

- L'URL complète ou la ligne de requête (par exemple, GET /path?param=value HTTP/1.1) ou
- L'en-tête Host combiné avec le chemin d'accès (par exemple, api.example.com/path)

Par conséquent, l'entrée regex pour HTTP peut contenir des caractères supplémentaires tels que /, ?, et des chaînes de requête, et pas seulement le nom d'hôte nu.

Implications de configuration

Un regex conçu uniquement pour les noms d'hôte (par exemple, seulement api.example.com correspondant) peut correspondre correctement à HTTPS (SNI) mais ne peut pas correspondre à la requête HTTP qui contient une URL complète ou une chaîne host+path.

Afin de filtrer le trafic HTTP et HTTPS avec le même modèle, vous devez :

- Modèles de conception principalement autour des noms d’hôte
- Vérifier le comportement vis-à-vis de HTTP et HTTPS dans les journaux UTD

Vérifier

Activer la consignation du débogage

Étape 1. Exécutez la commande debug utd engine standard url-filtering level info afin d'activer la journalisation de débogage.

Étape 2. Exécutez la commande show logging process ioxman module utd | include api.example.com afin de vérifier les journaux.

Exemple de rapport :

```
2025/11/27 11:45:28.195000350 {ioxman_R0-0}{255}: [utd] [21292]: (note): :(#0):INSP-URLF event->server_
2025/11/27 11:45:28.195001873 {ioxman_R0-0}{255}: [utd] [21292]: (note): :(#0):INSP-URLF   URL: api.exa
2025/11/27 11:45:28.195009216 {ioxman_R0-0}{255}: [utd] [21292]: (note): :(#0):INSP-URLF Regex matched
2025/11/27 11:45:28.195022442 {ioxman_R0-0}{255}: [utd] [21292]: (note): :(#0):INSP-URLF URLF whitelist
2025/11/27 11:45:33.530605572 {ioxman_R0-0}{255}: [utd] [21292]: (note): :(#0):INSP-URLF   URL: api.exa
2025/11/27 11:45:33.530606333 {ioxman_R0-0}{255}: [utd] [21292]: (note): :(#0):INSP-URLF Regex not matc
2025/11/27 11:45:33.530614980 {ioxman_R0-0}{255}: [utd] [21292]: (note): :(#0):INSP-URLF URLF whitelist
```

Exemples de configuration

Correspondance basée sur l'hôte

Afin d'autoriser tous les sous-domaines de example.com, utilisez ce modèle recommandé axé sur le nom d'hôte (ligne de base) :

```
^([a-zA-Z0-9-]+\.)*example\.com$
```

Ce modèle :

- Correspond à exemple.com, api.exemple.com, foo.bar.exemple.com, etc
- Convient à la correspondance HTTPS (SNI)
- Peut également correspondre à HTTP si la chaîne vue par le moteur est le nom d'hôte nu

Correspondance hôte/chemin HTTP

Si HTTP inclut host/path et que vous voulez ignorer le chemin, vous pouvez faire correspondre le préfixe de nom d'hôte et laisser l'expression rationnelle s'arrêter à une limite de mot au lieu d'une limite de fin. *, par exemple :

```
^([a-zA-Z0-9-]+\.)*example\.com\b
```

Remarque : Ici, \b (limite de mot) autorise efficacement les caractères tels que / ou ? afin de suivre le nom d'hôte sans nécessiter de caractère générique explicite .*. Ceci est généralement moins cher que l'ajout de .* à la fin et s'aligne mieux avec le guidage afin d'éviter des caractères génériques non limités supplémentaires.

Mise en garde : La chaîne exacte transmise au moteur d'expressions rationnelles pour les

⚠ requêtes HTTP est spécifique à l'implémentation et peut évoluer. En cas de doute, testez les modèles par rapport au trafic HTTP et HTTPS dans un environnement de travaux pratiques et vérifiez les correspondances dans les journaux UTD avant de les déployer en production.

Informations connexes

- Guide de configuration de la sécurité Cisco Catalyst SD-WAN, Cisco IOS XE Catalyst SD-WAN version 17.x
- Assistance technique de Cisco et téléchargements

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.