Richtlijnen voor reguliere expressies en prestatieoverwegingen voor URL-filtering

Inhoud

Inleiding

Dit document beschrijft de richtlijnen en prestatieoverwegingen voor het gebruik van reguliere expressies in URL-filtering met de UTD-engine. URL-filtering in de UTD-engine maakt gebruik van de PCRE2-bibliotheek voor reguliere expressies.

Bijgedragen door Eugene Khabarov, Cisco Engineering.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- Reguliere expressies (regex) syntaxis
- concepten voor URL-filtering
- Unified Threat Defense (UTD)-configuratie
- HTTPS/HTTP-protocolverschillen

Gebruikte componenten

Dit document is niet beperkt tot specifieke software- en hardware-versies.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

Hoewel PCRE2 krachtig is, kunnen bepaalde complexe of 'hebzuchtige' expressies buitensporige backtracking veroorzaken en interne limieten in de regex-engine raken. Wanneer dit gebeurt, kan een patroon te veel tijd in beslag nemen om te verwerken en uiteindelijk worden behandeld als 'geen match'.

Belangrijkste punten

- PCRE2 handhaaft interne limieten voor backtrackingstappen of matchtijd om systeembronnen te beschermen.
- Sommige patronen zijn syntactisch geldig, maar computationeel onveilig en kunnen 'catastrofale backtracking' veroorzaken.
- Wanneer deze limieten worden overschreden, kan de regex-engine de verwerking afbreken en geen overeenkomst retourneren, zelfs als de URL logisch overeenkomt met het patroon.

Patronen om te vermijden

Vermijd regex constructies die het volgende combineren:

- Geneste kwantificeerders, bijvoorbeeld: (...+)*, (.*)*, (.+)+, enzovoort
- Wildcards (.) herhaald over grote delen van de string, vooral in de buurt van het einde van het patroon
- Onontsnapte punten in domeinnamen bij gebruik in combinatie met herhaling

Hier is het patroon bijvoorbeeld syntactisch geldig, maar kan het duur zijn om te verwerken:

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

Opmerking: In dit geval is ([a-zA-Z0-9-]+.)* een groep met een geneste kwantificator (+ binnen *) plus een jokerteken (.). Op sommige niet-overeenkomende ingangen kan de regex-engine een zeer groot aantal backtrackingpaden verkennen.

Aanbevolen best practices

Altijd ontsnappingspunten in hostnamen

Gebruik \. om een letterlijke punt te matchen, bijvoorbeeld:

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

Ankerpatronen en tekens beperken

Gebruik ^ en $ en beperk tot verwachte tekens (bijvoorbeeld [a-zA-Z0-9-] voor hostlabels) om het terugtrekken te verminderen.

Vermijd geneste, onbegrensde herhaling waar mogelijk

Liever eenvoudigere constructies dan complexe patronen die alles in één retex proberen te bedekken. Overweeg verschillende specifieke vermeldingen in plaats van één zeer brede uitdrukking.

Testpatronen in een PCRE2-compatibele tester

Test vóór de implementatie regex-patronen in een PCRE2-compatibele omgeving en vermijd patronen die leiden tot 'catastrofale backtracking' of soortgelijke waarschuwingen.

Opmerking: Als een regex-patroon de interne limieten van de PCRE2-engine bereikt, kan het worden behandeld als 'geen overeenkomst' door de URL-filterengine. In dergelijke gevallen valt de URL-classificatie terug naar categorie of reputatie, niet naar het resultaat van de whitelist/blacklist regex. De exacte limieten zijn implementatiespecifiek en kunnen tussen releases veranderen. Je moet regexen conservatief ontwerpen.

Verschillen in URL Matching voor HTTP en HTTPS

De UTD-engine inspecteert URL's op verschillende manieren voor HTTPS- en HTTP-verkeer. Dit heeft invloed op hoe reguliere expressies moeten worden ontworpen voor URL-filtering.

HTTPS (TLS)-verkeer

Voor gecodeerd HTTPS-verkeer decodeert de UTD-engine de payload niet standaard.

- URL-filtering maakt gebruik van de Server Name Indication (SNI) van de Transport Layer Security (TLS) ClientHello.
- Het regex-patroon wordt alleen toegepast op de SNI-hostnaam, bijvoorbeeld: api.example.com

In dit geval wordt een hostname-gebaseerd patroon vergeleken met de hostname string api.example.com zoals:

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

HTTP-verkeer (niet-versleuteld)

Voor gewoon HTTP-verkeer kan de UTD-engine het volledige HTTP-verzoek zien (aanvraagregel en headers).

Afhankelijk van de implementatie kan de string die aan de regex engine wordt gegeven, het volgende bevatten:

- De volledige URL of verzoekregel (bijvoorbeeld GET /path?param=value HTTP/1.1) of
- De koptekst van de host wordt gecombineerd met het pad (bijvoorbeeld api.example.com/path)

Als gevolg hiervan kan de regex-invoer voor HTTP extra tekens bevatten zoals /, ? en querytekenreeksen, niet alleen de kale hostnaam.

Configuratie implicaties

Een regex die puur is ontworpen voor hostnamen (bijvoorbeeld alleen overeenkomende api.example.com) kan correct overeenkomen met HTTPS (SNI), maar kan niet overeenkomen met HTTP-verzoek dat een volledige URL of host+path-tekenreeks bevat.

Om zowel HTTP- als HTTPS-verkeer met hetzelfde patroon te filteren, moet u:

- Ontwerppatronen voornamelijk rond hostnamen
- Verifieer gedrag tegen zowel HTTP als HTTPS in de UTD-logs

Verifiëren

Debug-logboekregistratie inschakelen

Stap 1. Voer de standaard url-filtering level info commando uit om debug logging in te schakelen.

Stap 2. Voer het logboekproces show ioxman module utd | include api.example.com commando

uit om de logs te verifiëren.

Voorbeeld van uitvoer:

```
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
```

Configuratievoorbeelden

Hostgebaseerde matching

Om alle subdomeinen van example.com toe te staan, gebruikt u dit aanbevolen hostname-gefocuste patroon (baseline):

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

Dit patroon:

- Komt overeen met example.com, api.example.com, foo.bar.example.com, enzovoort
- Geschikt voor HTTPS (SNI) matching
- Kan ook overeenkomen met HTTP als de string die door de engine wordt gezien de kale hostnaam is

HTTP-host/overeenkomend pad

Als HTTP host/path bevat en u het pad wilt negeren, kunt u het voorvoegsel voor de hostnaam matchen en de regex laten stoppen bij een woordgrens in plaats van een achtervoegsel. * bijvoorbeeld:

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

Opmerking: Hier staat \b (woordgrens) effectief tekens toe zoals /of ? om de hostnaam te volgen zonder een expliciete .* jokerteken te vereisen. Dit is over het algemeen goedkoper dan toevoegen .* aan het einde en sluit beter aan bij de begeleiding om extra onbegrensde wildcards te voorkomen.

⚠ Let op: de exacte string die in de regex engine wordt doorgegeven voor HTTP-verzoeken is implementatiespecifiek en kan evolueren. Test bij twijfel patronen tegen zowel HTTP- als HTTPS-verkeer in een laboratoriumomgeving en verifieer overeenkomsten in de UTD-logs voordat u deze implementeert voor productie.

Gerelateerde informatie

- Cisco Catalyst SD-WAN Security Configuration Guide, Cisco IOS XE Catalyst SD-WAN Release 17.x
- Cisco Technical Support en downloads

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.