URL過濾的正規表示式准則和效能注意事項

目錄

簡介

本文檔介紹在UTD引擎的URL過濾中使用正規表示式的指導原則和效能注意事項。UTD引擎中的URL過濾使用PCRE2正規表示式庫。

作者：Eugene Khabarov，思科工程公司。

必要條件

需求

思科建議您瞭解以下主題：

- 正規表示式(regex)語法
- URL過濾概念
- 整合威脅防禦(UTD)組態

- HTTPS/HTTP協定差異

採用元件

本文件所述內容不限於特定軟體和硬體版本。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

雖然PCRE2功能強大，但某些複雜或「貪婪」表達式會導致過度回溯，並可能觸及regex引擎的內部限制。如果發生這種情況，處理模式可能會花費太多時間，最終會視為「無匹配」。

要點

- PCRE2對回溯步驟或匹配時間實施內部限制，以保護系統資源。
- 某些模式在語法上有效，但在計算上不安全，並且可能觸發「災難性回溯」。
- 超出這些限制時，即使該URL在邏輯上與該模式匹配，正規表示式引擎也可能中止處理並且不返回匹配項。

要避免的模式

避免組合以下內容的regex結構：

- 巢狀的限定符，例如：(...+)*、(.*)*、(.+)+等等
- 萬用字元(.)在字串的大部分上重複，特別是在模式末尾附近
- 與重複一起使用時，域名中的未轉義點

例如，這裡的模式在語法上是有效的，但處理起來可能很昂貴：

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

附註：在本例中， ([a-zA-Z0-9-]+.)*是一個帶有巢狀量化符(+inside *)加上萬用字元(.)的組。在一些非匹配輸入上，regex引擎可以探索大量回溯路徑。

建議的最佳實踐

主機名中的始終轉義點

使用\.以匹配文字點，例如：

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

錨點模式和限制字元

使用^和$並限製為預期字元(例如，對於主機標籤，使用[a-zA-Z0-9-])，以減少回溯。

儘可能避免巢狀的無限重複

選擇更簡單的結構，而不是試圖在一個正規表示式中涵蓋所有內容的複雜模式。考慮幾個特定的條目，而不是一個非常寬泛的表達式。

PCRE2相容測試儀中的測試模式

在部署之前，請在與PCRE2相容的環境中測試regex模式，並避免出現「災難性回溯」或類似警告的模式。

附註：如果正規表示式模式達到PCRE2引擎的內部限制，則URL過濾引擎會將其視為「無匹配」。在這種情況下，URL分類將回退到類別或信譽，而不是白名單/黑名單正規表示式結果。確切的限制因實施而異，可能會在版本之間更改。你必須保守地設計正規表示式。

HTTP和HTTPS的URL匹配差異

UTD引擎會針對HTTPS和HTTP流量以不同的方式檢查URL。這會影響針對URL過濾設計正規表示式的方式。

HTTPS(TLS)流量

對於加密的HTTPS流量，預設情況下UTD引擎不會解密負載。

- URL過濾使用來自傳輸層安全(TLS)客戶端Hello的伺服器名稱指示(SNI)。
- 正規表示式模式僅應用於SNI主機名，例如：api.example.com

在這種情況下，將基於主機名的模式與主機名字串api.example.com進行匹配，例如：

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

HTTP（未加密）流量

對於普通HTTP流量，UTD引擎可以看到完整的HTTP請求（請求行和標頭）。

根據實施情況，為regex引擎提供的字串可能包括：

- 完整的URL或請求行(例如GET /path?param=value HTTP/1.1)或

- 主機標頭與路徑相結合(例如api.example.com/path)

因此，HTTP的regex輸入可以包含附加字元(如/、?)和查詢字串，而不僅僅是裸主機名。

配置影響

完全為主機名設計的正規表示式(例如，僅匹配api.example.com)可以正確匹配HTTPS(SNI)，但無法匹配包含完整URL或主機+路徑字串的HTTP請求。

若要以相同模式篩選HTTP和HTTPS流量，您必須：

- 主要圍繞主機名設計模式
- 驗證UTD日誌中針對HTTP和HTTPS的行為

驗證

啟用調試日誌記錄

步驟1.運行debug utd engine standard url-filtering level info命令以啟用調試日誌記錄。

步驟2.運行show logging process ioxman module utd | include api.example.com命令以驗證日誌。

輸出示例：

```
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
```

組態設定範例

基於主機的匹配

若要允許example.com的所有子域，請使用以下推薦的以主機名為中心的模式（基線）：

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

此模式：

- 匹配example.com、api.example.com、foo.bar.example.com等

- 適用於HTTPS(SNI)配對
- 如果引擎看到的字串是裸主機名，也可以匹配HTTP

HTTP主機/路徑匹配

如果HTTP包含主機/路徑並且您想要忽略該路徑，則可以匹配主機名字首並讓正規表示式停止在字邊界而不是尾部。*例如：

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

附註：這裡，\b（字元邊界）實際上允許使用字元，例如/或?，以便跟隨主機名稱，而不需要顯式。*萬用字元。這通常比新增便宜。*在結尾處與指南更好地對齊，以避免新增無界萬用字元。

注意：針對HTTP請求傳遞到regex引擎的確切字串是特定於實現的，並且可以演變。如果有疑問，請在實驗室環境中對HTTP和HTTPS流量測試模式，並在部署到生產環境之前驗證UTD日誌中的匹配項。

相關資訊

- Cisco Catalyst SD-WAN安全配置指南，Cisco IOS XE Catalyst SD-WAN版本17.x
- 思科技術支援與下載

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。