URL过滤的正则表达式指南和性能注意事项

目录

简介

本文档介绍在UTD引擎的URL过滤中使用正则表达式的指导原则和性能注意事项。UTD引擎中的URL过滤使用PCRE2正则表达式库。

作者：Eugene Khabarov，思科工程部。

先决条件

要求

Cisco 建议您了解以下主题：

- 正则表达式(regex)语法
- URL过滤概念
- 统一威胁防御(UTD)配置

- HTTPS/HTTP协议差异

使用的组件

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

虽然PCRE2功能强大，但某些复杂或“贪婪”表达式可能导致过度回溯，并可能达到regex引擎的内部限制。发生这种情况时，处理模式可能花费太多时间，最终会视为“无匹配”。

要点

- PCRE2对回溯步骤或匹配时间实施内部限制，以保护系统资源。
- 某些模式在语法上是有效的，但在计算上是不安全的，并且可能触发“灾难性回溯”。
- 当超过这些限制时，regex引擎可以中止处理并且不返回匹配项，即使URL在逻辑上与模式匹配。

要避免的模式

避免结合以下内容的regex结构：

- 嵌套的限定符，例如：(...+)*、(.*)*、(.+)+等等
- 通配符(.)在字符串的大部分上重复，特别是在模式末尾附近
- 与重复一起使用时，域名中的非转义点

例如，这里的模式在语法上是有效的，但处理起来可能很昂贵：

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

注意：在本例中，([a-zA-Z0-9-]+.)*是一个带有嵌套量词(+ inside *)加上通配符(.)的组。 在一些非匹配输入上，regex引擎可以探索大量回溯路径。

建议的最佳实践

主机名中的始终转义点

使用\.以匹配文字点，例如：

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

锚模式和限制字符

使用^和$并限制为预期字符(例如，对于主机标签，使用[a-zA-Z0-9-])以减少回溯。

尽可能避免嵌套、无限制的重复

更喜欢简单的结构，而不是试图在一个regex中涵盖所有内容的复杂模式。请考虑几个特定的条目，而不是一个非常宽泛的表达式。

在兼容PCRE2的测试器中测试模式

在部署之前，请在兼容PCRE2的环境中测试regex模式，并避免出现“灾难性回溯”或类似警告的模式。

注意：如果正则表达式模式达到PCRE2引擎的内部限制，URL过滤引擎可将其视为“无匹配”。在这种情况下，URL分类将退回到类别或信誉，而不是白名单/黑名单正则表达式结果。确切限制因实施而异，可能会在版本之间更改。您必须保守地设计正则表达式。

HTTP和HTTPS的URL匹配差异

UTD引擎对HTTPS和HTTP流量以不同方式检查URL。这会影响必须为URL过滤设计正则表达式的方式。

HTTPS(TLS)流量

对于加密的HTTPS流量，默认情况下，UTD引擎不会解密负载。

- URL过滤使用来自传输层安全(TLS)客户端Hello的服务器名称指示(SNI)。
- regex模式仅应用于SNI主机名，例如：api.example.com

在这种情况下，基于主机名的模式与主机名字符串api.example.com匹配，例如：

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

HTTP（未加密）流量

对于普通的HTTP流量，UTD引擎可以看到完整的HTTP请求（请求行和报头）。

根据实施情况，指定给regex引擎的字符串可能包括：

- 完整的URL或请求行(例如，GET /path?param=value HTTP/1.1)或

- 主机报头与路径相结合(例如api.example.com/path)

因此，HTTP的regex输入可以包含其他字符，例如/、?和查询字符串，而不仅仅是裸主机名。

配置影响

完全为主机名设计的regex(例如，仅匹配api.example.com)可以正确匹配HTTPS(SNI)，但无法匹配包含完整URL或host+path字符串的HTTP请求。

要以相同模式过滤HTTP和HTTPS流量，您必须：

- 主要围绕主机名设计模式
- 验证UTD日志中针对HTTP和HTTPS的行为

验证

启用调试日志记录

步骤1.运行debug utd engine standard url-filtering level info命令以启用调试日志记录。

步骤2.运行show logging process ioxman module utd | include api.example.com命令以验证日志。

示例输出：

```
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
```

配置示例

基于主机的匹配

要允许example.com的所有子域，请使用以下推荐的以主机名为中心的模式（基线）：

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

此模式：

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

- 适用于HTTPS(SNI)匹配
- 如果引擎看到的字符串是裸主机名，也可以匹配HTTP

HTTP主机/路径匹配

如果HTTP包含主机/路径，而您想要忽略该路径，则可以匹配主机名前缀，并让regex停止在字边界而不是尾部。*例如：

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

注意：此处，\b（字边界）有效地允许字符(如/或?)以便跟随主机名，而不需要显式。*通配符。这通常比添加.*在末尾要便宜，并且与指南更一致，以避免添加额外的无界通配符。

警告：为HTTP请求传递到regex引擎的确切字符串是特定于实现的，并且可以演进。如有疑问，请在实验环境中针对HTTP和HTTPS流量测试模式，并在部署到生产环境之前验证UTD日志中的匹配项。

相关信息

- Cisco Catalyst SD-WAN安全配置指南，Cisco IOS XE Catalyst SD-WAN版本17.x
- 思科技术支持和下载

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。