Diretrizes de expressões regulares e considerações de desempenho para filtragem de URL

Contents

Introdução

Este documento descreve as diretrizes e considerações de desempenho para o uso de expressões regulares na filtragem de URL com o mecanismo UTD. A filtragem de URL no mecanismo UTD usa a biblioteca de expressões regulares do PCRE2.

Contribuição de Eugene Khabarov, Cisco Engineering.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Sintaxe de expressões regulares (regex)
- Conceitos de filtragem de URL
- Configuração do Unified Threat Defense (UTD)
- Diferenças de protocolo HTTPS/HTTP

Componentes Utilizados

Este documento não se restringe a versões de software e hardware específicas.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

Embora o PCRE2 seja poderoso, certas expressões complexas ou "gananciosas" podem causar retrocesso excessivo e atingir limites internos no mecanismo regex. Quando isso ocorre, um padrão pode levar muito tempo para ser processado e, por fim, ser tratado como "sem correspondência".

Pontos principais

- O PCRE2 impõe limites internos nas etapas de backtracking ou no tempo de correspondência para proteger os recursos do sistema.
- Alguns padrões são sintaticamente válidos, mas computacionalmente inseguros e podem disparar "backtracking catastrófico".
- Quando esses limites são excedidos, o mecanismo regex pode abortar o processamento e não retornar nenhuma correspondência, mesmo que o URL corresponda logicamente ao padrão.

Padrões para evitar

Evite construções regex que combinem:

- Quantificadores aninhados, por exemplo: (...+)*, (.*)*, (.+)+ e assim por diante
- Caracteres curinga (.) repetidos em grandes partes da string, especialmente perto do final do padrão
- Pontos sem escape em nomes de domínio quando usados com repetição

Por exemplo, aqui o padrão é sintaticamente válido, mas pode ser caro de processar:

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

Note: Nesse caso, ([a-zA-Z0-9-]+.)* é um grupo com um quantificador aninhado (+ dentro de *) mais um curinga (.). Em algumas entradas não correspondentes, o mecanismo regex pode explorar um número muito grande de caminhos de backtracking.

Práticas recomendadas

Sempre Escapar Pontos em Nomes de Host

Use \. para corresponder um ponto literal, por exemplo:

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

Padrões de âncora e caracteres restritos

Use ^ e $ e restrinja aos caracteres esperados (por exemplo, [a-zA-Z0-9-] para rótulos de host) para reduzir o backtracking.

Evite Repetição Aninhada E Ilimitada Quando Possível

Preferem construções mais simples em vez de padrões complexos que tentam cobrir tudo em um regex. Considere várias entradas específicas em vez de uma expressão muito ampla.

Testar padrões em um testador compatível com PCRE2

Antes da implantação, teste os padrões regex em um ambiente compatível com PCRE2 e evite padrões que gerem 'backtracking catastrófico' ou avisos semelhantes.

Note: Se um padrão regex atingir os limites internos do mecanismo PCRE2, ele poderá ser tratado como 'sem correspondência' pelo mecanismo de filtragem de URL. Nesses casos, a classificação de URL volta para a categoria ou reputação, e não para o resultado do regex da lista branca/lista negra. Os limites exatos são específicos de implementação e podem mudar entre versões. Você deve projetar regexes de forma conservadora.

Diferenças na correspondência de URL para HTTP e HTTPS

O mecanismo UTD inspeciona URLs de forma diferente para tráfego HTTPS e HTTP. Isso afeta como as expressões regulares devem ser projetadas para a filtragem de URL.

Tráfego HTTPS (TLS)

Para tráfego HTTPS criptografado, o mecanismo UTD não descriptografa o payload por padrão.

- A filtragem de URL usa a SNI (Server Name Indication, indicação de nome de servidor) do TLS ClientHello.
- O padrão regex é aplicado somente ao nome de host SNI, por exemplo: api.example.com

Nesse caso, um padrão baseado em nome de host é comparado com a string de nome de host api.example.com como:

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

Tráfego HTTP (não criptografado)

Para tráfego HTTP simples, o mecanismo UTD pode ver a solicitação HTTP completa (linha de solicitação e cabeçalhos).

Dependendo da implementação, a string fornecida ao mecanismo regex pode incluir:

- A URL completa ou a linha de solicitação (por exemplo, GET /path?param=value HTTP/1.1) ou
- O cabeçalho do Host combinado com o caminho (por exemplo, api.example.com/path)

Como resultado, a entrada regex para HTTP pode conter caracteres adicionais como /, ? e strings de consulta, não apenas o nome de host puro.

Implicações da configuração

Um regex projetado puramente para nomes de host (por exemplo, apenas api.example.com correspondente) pode corresponder corretamente ao HTTPS (SNI), mas não corresponde à solicitação HTTP que contém uma URL completa ou uma string host+caminho.

Para filtrar o tráfego HTTP e HTTPS com o mesmo padrão, você deve:

- Padrões de design principalmente em torno de nomes de host
- Verificar o comportamento em relação a HTTP e HTTPS nos logs UTD

Verificar

Habilitar Log de Depuração

Etapa 1. Execute o comando debug utd engine standard url-filtering level info para habilitar o registro de depuração.

Etapa 2. Executar o comando show logging process ioxman module utd | include api.example.com para verificar os registros.

Saída de exemplo:

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

Exemplos de configuração

Correspondência baseada em host

Para permitir todos os subdomínios de example.com, use este padrão focado no nome de host recomendado (linha de base):

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

Este padrão:

- Corresponde a example.com, api.example.com, foo.bar.example.com e assim por diante
- É adequado para correspondência HTTPS (SNI)
- Também pode corresponder ao HTTP se a string vista pelo mecanismo for o nome do host vazio

Correspondência de Host/Caminho HTTP

Se o HTTP incluir host/caminho e você quiser ignorar o caminho, poderá corresponder o prefixo do nome do host e deixar que o regex pare em um limite de palavra em vez de no final. *, por exemplo:

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

Note: Aqui, \b (limite de palavra) permite efetivamente caracteres como / ou ? para seguir o nome de host sem exigir um curinga .* explícito. Isso geralmente é mais barato do que adicionar .* no final e se alinha melhor com a orientação, a fim de evitar curingas não-limitados adicionais.

Caution: A sequência de caracteres exata passada para o mecanismo regex para solicitações HTTP é específica de implementação e pode evoluir. Em caso de dúvida, teste padrões em relação ao tráfego HTTP e HTTPS em um ambiente de laboratório e verifique

⚠ as correspondências nos logs UTD antes de implantar na produção.

Informações Relacionadas

- Guia de configuração de segurança do Cisco Catalyst SD-WAN, Cisco IOS XE Catalyst SD-WAN versão 17.x
- Suporte técnico e downloads da Cisco

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.