Konfigurieren und Validieren regulärer Ausdrücke in Cisco ESA und CES

Inhalt

Einleitung

In diesem Dokument wird beschrieben, wie die ESA und die CES reguläre Ausdrücke in Filtern verwenden, wie sich ihr Verhalten unterscheidet und wie sie vor ihrer Durchsetzung getestet werden müssen.

Hintergrundinformationen

In diesem Dokument wird beschrieben, wie die Cisco E-Mail Security Appliance (ESA) und Cisco Cloud E-Mail Security (CES) reguläre Ausdrücke behandeln, wenn sie in Nachrichtenfiltern und Content-Filtern verwendet werden. Der Schwerpunkt liegt dabei auf dem Verständnis, wie sich reguläre Ausdrücke in diesen Komponenten verhalten und wie sie mit E-Mail-Headern, Body Content und Anhängen interagieren.

Es ist wichtig, von Anfang an klarzustellen, dass sich das im SvD-Modul verwendete Modul für reguläre Ausdrücke anders verhält. Daher gilt alles, was in diesem Dokument beschrieben wird, ausschließlich für Nachrichtenfilter und Content-Filter und nicht für SvD-Richtlinien.

Wenn Administratoren mit regulären Ausdrücken in ESA arbeiten, müssen sie verstehen, dass E-Mail-Inhalte nicht auf die gleiche Weise ausgewertet werden wie in einem E-Mail-Client. E-Mail-Nachrichten enthalten Envelope-Informationen, strukturierte Header, MIME-Komponenten und potenziell codierten Inhalt. Daher können Vergleiche, die von Filtern durchgeführt werden, zu unerwarteten Ergebnissen führen, wenn die Nachrichtenstruktur und das Regex-Verhalten nicht vollständig verstanden werden.

Aus diesem Grund können neue Filter, die reguläre Ausdrücke verwenden, immer im Überwachungsmodus aktiviert werden, bevor sie durchgesetzt werden. Dies ermöglicht die Validierung gegen echten Datenverkehr und verhindert unbeabsichtigte Blockierungen oder Leistungseinbußen.

Wörterbücher und Suchbegriffe

Beim Erstellen eines Nachrichtenfilters oder eines Content-Filters wird der in vielen Bedingungen eingegebene Begriff als regulärer Ausdruck interpretiert. Dies ist ein kritisches Konzept: Selbst wenn der Administrator beabsichtigt, den Text mit einem Literal zu vergleichen, kann die ESA die Eingabe mithilfe der Regex-Logik verarbeiten.

Dies gilt nicht für alle Bedingungstypen. Wenn Sie beispielsweise unter bestimmten strukturierten Bedingungen nach einer bestimmten IP-Adresse suchen, wird der Wert nicht als regulärer Ausdruck interpretiert. Beim Suchen im Betreff-Header, im Nachrichtentext, in einem bestimmten Headerfeld oder in einem Dateinamen einer Anlage wird der Wert jedoch in der Regel als reguläres Muster behandelt.

Ein typisches Beispiel veranschaulicht dies deutlich. Angenommen, das Ziel besteht darin, E-Mails mit dem Betreff zu blockieren:

```
Receipt number (123456)
```

Da Klammern Sonderzeichen in regulären Ausdrücken sind (die für die Gruppierung verwendet werden), müssen sie mit Escapezeichen versehen werden.

Der richtige Ausdruck wäre:

```
Receipt number \(123456\)
```

Wenn die Klammern nicht mit Escapezeichen versehen sind, interpretiert das Regex-Modul sie als Gruppierungsoperatoren und nicht als Literalzeichen. Je nach Muster kann dies unbeabsichtigte Übereinstimmungen oder ein anderes Verhalten als erwartet verursachen.

Aus diesem Grund ist es wichtig zu verstehen, welche Zeichen eine spezielle Bedeutung in regex haben und sicherzustellen, dass sie richtig entkommen, wenn Literalvergleich erforderlich ist.

Beispiele für Sonderzeichen und ihre Escapesyntax

Die erste Spalte zeigt einen Beispieltext mit Sonderzeichen, und die zweite Spalte zeigt, wie die richtige Syntax für reguläre Ausdrücke geschrieben werden muss, damit sie mit dem Literaltext in Cisco ESA (Python-Stil regex) übereinstimmt.

Passender Literaltext	Richtige Syntax für reguläre Ausdrücke
Belegnummer (123456)	Belegnummer \(123456\)
user@example.com	user@example\.com
www.test.abc	www\.test\.abc
file_name.txt	file_name\.txt
Preis ist 10,50	Preis ist 10\.50
C:\Users\Admin	C:\\Users\\Admin
[VERTRAULICH]	\[VERTRAULICH\]
{Rechnung}	\{Rechnung\}
+34 600 123 456	\+34 600 123 456
Frage?	Frage\?
100% garantiert	100% garantiert (% muss nicht entkommen)
Sternchen * Symbol	Sternchen * Symbol
A|B	A|B
Caret ^Anfang	Caret \^start
100 USD	Dollar \$100

Einschränkung der Verwendung regulärer Ausdrücke

Reguläre Ausdrücke müssen vorsichtig und nur bei Bedarf verwendet werden. Obwohl sie leistungsstarke Übereinstimmungsfunktionen bieten, können übermäßige oder schlecht konzipierte Ausdrücke die Verarbeitungszeit für Nachrichten verlängern und zu unbeabsichtigten Übereinstimmungen führen.

Ein bestimmtes Konstrukt, das Vorsicht erfordert, ist .* , das "ein beliebiges Zeichen, null oder mehr Male" repräsentiert. Wenn sie am Anfang oder Ende eines Ausdrucks platziert wird, kann sie zu einem übermäßigen Rücklauf und unnötigem Verarbeitungsaufwand führen.

Die Cisco Dokumentation zeigt an, dass Einträge, die .* am Anfang oder Ende verwenden, das System unter bestimmten Bedingungen sperren können, wenn bestimmte MIME-Teile zugeordnet werden. Aus diesem Grund empfiehlt Cisco, die Verwendung von voran- oder nachgestellten .* zu vermeiden, wann immer dies möglich ist.

In vielen Szenarien verwenden Administratoren Muster wie .*bill.*, wenn sie Rechnungen einfach schreiben und das gleiche praktische Ergebnis in der ESA erzielen konnten. Da die Scan-Engine bereits die relevanten Inhaltsbereiche durchsucht, ist das Umschließen eines Wortes mit .* häufig redundant und rechnerisch ineffizient.

⚠ Vorsicht: Generell wird empfohlen, reguläre Ausdrücke so einfach und präzise wie möglich zu halten.

Nachrichtenfilter, Content-Filter und Wörterbücher

Die Cisco ESA bietet mehrere Mechanismen zur Auswertung von Meldungen und zum Anwenden von Aktionen. Nachrichtenfilter arbeiten am Anfang der Pipeline und verwenden eine Skriptsyntax. Sie sind äußerst flexibel und ermöglichen eine erweiterte Logik, die Umschlagdaten, Header und Anhangseigenschaften umfasst. Da sie jedoch früh in der Verarbeitungskette ausgeführt werden, können ineffiziente Nachrichtenfilter die Leistung beeinträchtigen.

Content-Filter werden über die grafische Benutzeroberfläche konfiguriert und funktionieren, nachdem die Nachricht akzeptiert wurde. Bei den meisten Anwendungsfällen der Inhaltsanalyse sind die Content-Filter einfacher zu verwalten und hinsichtlich der Leistung sicherer.

Sowohl bei den Nachrichtenfiltern als auch bei den Inhaltsfiltern können reguläre Ausdrücke entweder direkt in einer Bedingung oder indirekt durch die Verwendung von Wörterbüchern eingeführt werden.

Mit Wörterbüchern können Administratoren wiederverwendbare Suchbegriffe zentralisieren. Jeder Eintrag wird in einer separaten Zeile geschrieben und kann als Nur-Text oder regulärer Ausdruck dargestellt werden. Wörterbücher unterstützen auch Nicht-ASCII-Zeichen und eignen sich daher für mehrsprachige Umgebungen.

In einigen Situationen können sich bestimmte komplexe Konstrukte für reguläre Ausdrücke in Wörterbüchern nicht identisch verhalten. In diesem Fall muss der reguläre Ausdruck direkt in der Filterbedingung anstatt im Wörterbuch platziert werden.

Die Cisco ESA ermöglicht die Erstellung von bis zu 150 Content-Dictionarys. Standardmäßig können 100 Wörterbücher konfiguriert werden, es sei denn, der Grenzwert wird mithilfe des Befehls dictionaryconfig über die CLI geändert.

Wörterbücher können auch die Gewichtung von Begriffen implementieren. Jedem Begriff kann eine numerische Gewichtung zugewiesen werden. Wenn die ESA eine Nachricht scannt, multipliziert sie die Anzahl der Vorkommen dieses Begriffs mit der Gewichtung. Das Ergebnis wird mit einem im Filter definierten Grenzwert verglichen. Dieses Bewertungsmodell ermöglicht eine flexiblere und abgestufte Durchsetzung von Richtlinien.

Darüber hinaus können Wörterbücher Smart Identifier enthalten, die algorithmische Detektoren für strukturierte numerische Muster wie Sozialversicherungsnummern oder Bankkennungen sind.

Regulärer Ausdruck-Engine

Die Cisco ESA verwendet reguläre Ausdrücke, die auf dem Python re-Modulstil basieren. Obwohl dies Kompatibilität mit gängiger Python-Regex-Syntax bietet, wird nicht jede erweiterte Funktion, die in vollständigen Python-Umgebungen unterstützt wird, notwendigerweise von der ESA unterstützt.

Für eine exakte Zeichenfolgenübereinstimmung müssen Ausdrücke mit ^ am Anfang und $ am Ende verankert werden. Ohne diese Anker kann die Regex-Engine Teilzeichenfolgen und nicht vollständige Werte zuordnen.

Der Ausdruck:

```
sun.com
```

Zeichenfolgen zuordnen, z. B.:

```
thegodsunocommando
```

Der Ausdruck:

```
^sun\.com$
```

Ordnen Sie nur die exakte Zeichenfolge sun.com zu.

Wenn Sie eine leere Zeichenfolge suchen, ist es wichtig, "" nicht zu verwenden, da dies effektiv mit allen Zeichenfolgen übereinstimmt. Stattdessen lautet der richtige Ausdruck:

```
^$
```

Da die Cisco ESA reguläre Ausdrücke im Python-Stil verwendet, gibt es eine Reihe von Möglichkeiten, einen Vergleich ohne Berücksichtigung der Groß-/Kleinschreibung durchzuführen.

Wie bereits erwähnt, wird bei regulären Ausdrücken standardmäßig die Groß- und Kleinschreibung berücksichtigt. Das heißt, Sie suchen nach:

```
foo
```

Nur passend für foo, aber nicht FOO, Foo oder Foo.

Wenn Sie eine Übereinstimmung ohne Berücksichtigung der Groß-/Kleinschreibung durchführen möchten, können Sie das Inline-Flag (?i) am Anfang des regulären Ausdrucks verwenden. Dies weist die Regex-Engine an, die Groß- und Kleinschreibung für den Rest des Musters zu ignorieren.

Beispiele:

```
(?i)foo
```

Dieser Ausdruck stimmt überein:

- Narr
- FOO
- Foo
- Aua

Wenn Sie die gesamte Zeichenfolge genau abgleichen möchten, ohne die Groß-/Kleinschreibung zu berücksichtigen, können Sie das Flag ohne Berücksichtigung der Groß-/Kleinschreibung mit Ankern kombinieren:

```
(?i)^foo$
```

Dadurch wird sichergestellt, dass der volle Wert unabhängig von der Groß- und Kleinschreibung genau "foo" ist.

Eine andere (weniger praktische) Alternative wäre die explizite Definition aller möglichen Kombinationen mithilfe von Zeichenklassen, z. B.:

```
[Ff][Oo][Oo]
```

Dieser Ansatz ist jedoch schwer zu pflegen und wird nicht empfohlen, wenn stattdessen das Flag (?i) verwendet werden kann.

In den meisten ESA-Szenarien ist die bevorzugte und sauberste Methode für einen Abgleich ohne Berücksichtigung der Groß-/Kleinschreibung:

```
(?i)
```

am Anfang des regulären Ausdrucks.

Nicht-ASCII-Zeichen und Wortgrenzen

In Sprachen, die Doppelbyte-Zeichensätze verwenden, können sich die Begriffe Wortgrenzen oder Groß-/Kleinschreibung nicht wie erwartet verhalten. Komplexe Ausdrücke, die von Konstrukten wie \w abhängen, können zu inkonsistenten Ergebnissen führen, wenn die Codierung oder das Gebietsschema unbekannt sind.

In solchen Fällen kann es ratsam sein, die Durchsetzung von Wortgrenzen in der Wörterbuchkonfiguration zu deaktivieren oder den Ausdruck zu vereinfachen, um die Abhängigkeit von mehrdeutigen Zeichenklassen zu vermeiden.

Bei Verwendung von Nicht-ASCII-Wörterbüchern kann die CLI-Anzeige Zeichen je nach Terminal-Kodierung nicht korrekt wiedergeben. In diesen Fällen wird empfohlen, das Wörterbuch in eine Textdatei zu exportieren, extern zu bearbeiten und erneut zu importieren.

Schreiben effizienter Filter

Die Effizienz ist beim Schreiben von Filtern besonders in Umgebungen mit hohen Datenvolumen von entscheidender Bedeutung. Ein häufiger Fehler besteht darin, lange Ketten von OR-Bedingungen für ähnliche Übereinstimmungen zu schreiben.

Wenn Sie beispielsweise Dutzende von Anbauverlängerungen einzeln prüfen, wird die Regex-Engine wiederholt initialisiert. Dies erhöht die CPU-Auslastung und verringert die Wartungsfreundlichkeit.

Anstatt viele verschiedene Vergleiche zu schreiben, reduziert das Gruppieren dieser Vergleiche mithilfe von Wechseln innerhalb eines einzelnen regulären Ausdrucks den Verarbeitungsaufwand erheblich. Dadurch wird die Anzahl der Aufrufe der Regex-Engine reduziert und die Wartung des Filters vereinfacht.

Ein effizientes Filterdesign bedeutet nicht nur, lesbar zu sein, sondern es wirkt sich auch direkt auf die Systemleistung aus.

PDFs und reguläre Ausdrücke

Die Zuordnung von Inhalten in PDF-Dateien kann je nach Art der Erstellung zu unerwarteten Ergebnissen führen. Einige PDF-Dateien enthalten in ihrer internen Darstellung keine logischen Leerzeichen oder Zeilenumbrüche. Die Scan-Engine versucht, den logischen Abstand basierend auf der Wortpositionierung zu rekonstruieren.

Wenn ein Wort mit mehreren Schriftarten oder Schriftgrößen erstellt wird, kann die interne Darstellung den Text fragmentieren. Beispielsweise kann das Wort "Legende" intern als "Aufruf" oder "c a l Legende" interpretiert werden.

In solchen Fällen kann der Versuch, den Ausdruck "callout" zu finden, fehlschlagen, da die interne Darstellung keine genau zusammenhängende Zeichenfolge enthält. Administratoren müssen sich dieser Einschränkung bewusst sein, wenn sie inhaltsbasierte Richtlinien für PDF-Anhänge entwerfen.

Testen regulärer Ausdrücke

Das Testen regulärer Ausdrücke vor ihrer Bereitstellung in der Produktionsumgebung ist eine wichtige betriebliche Anforderung. Ein regulärer Ausdruck, der syntaktisch korrekt erscheint, kann sich in Bezug auf den tatsächlichen E-Mail-Verkehr sehr unterschiedlich verhalten. Ohne ordnungsgemäße Tests kann ein Filter Fehlalarme generieren, unbeabsichtigte Muster nicht erkennen, einen Performance-Overhead verursachen oder unbeabsichtigt den legitimen E-Mail-Fluss unterbrechen.

Die Prüfung muss als strukturierter, zweistufiger Prozess angegangen werden, um das Risiko zu minimieren, bevor ein Filter in der Produktion aktiviert wird.

Phase 1 - Entwurf und Validierung regulärer Ausdrücke
Der Schwerpunkt der ersten Phase liegt auf der Entwicklung und Validierung des regulären Ausdrucks selbst, bevor dieser in die Cisco ESA integriert wird.

1. Verwendung von regex101 oder ähnlichen Werkzeugen

Online-Plattformen wie http://regex101.com (oder gleichwertige Tools) sind während der Entwurfsphase äußerst nützlich. Bei Verwendung dieser Tools muss die Python-Variante ausgewählt werden, um die ESA-Regex-Engine anzunähern.

Diese Plattformen ermöglichen Administratoren Folgendes:

- Validierung der korrekten Syntax
- Bestätigen, dass Sonderzeichen ordnungsgemäß mit Escapezeichen versehen sind
- Testen von übereinstimmenden und nicht übereinstimmenden Fällen
- Gruppierungs- und Quantifiziererverhalten visualisieren
- Identifizierung potenziell gieriger Konstrukte wie .*

Diese Tools simulieren jedoch das standardmäßige Python-regulatorische Verhalten und können Funktionen unterstützen, die nicht vollständig in die Cisco ESA implementiert sind. Daher müssen sie als vorläufige Validierungstools und nicht als endgültige Kompatibilitätstests betrachtet werden.

2. Verwendung von KI-Modellen (ChatGPT, Copilot, ...)

KI-basierte Assistenten können die Erstellung von regulären Ausdrücken beschleunigen, insbesondere bei komplexen Übereinstimmungsszenarien. Durch die Beschreibung des gewünschten Verhaltens in natürlicher Sprache können Administratoren einen ersten Vorschlag für einen regulären Ausweis erhalten, der dann verfeinert werden kann.

KI-Tools sind besonders hilfreich bei:

- Generieren komplexer gruppierter Ausdrücke
- Konvertieren von Geschäftsanforderungen in eine reguläre Syntax
- Vereinfachung langer ODER-basierter Bedingungen in gruppierten Wechseln

Dennoch müssen KI-generierte Ausdrücke immer kritisch überprüft werden. Sie können Ineffizienzen, nicht unterstützte Konstrukte oder übermäßig komplexe Logik einführen. Die KI-Unterstützung muss als Entwurfsbeihilfe und nicht als endgültige Validierung behandelt werden. Jeder KI-generierte Ausdruck muss noch mit strukturierten Validierungsmethoden getestet werden.

Phase 2 - Überprüfung des Filterverhaltens mit der Cisco ESA

Nachdem der Ausdruck selbst validiert wurde, konzentriert sich die zweite Phase darauf, zu bestätigen, wie er sich innerhalb der Cisco ESA verhält, wenn er auf die Verarbeitung echter Nachrichten angewendet wird.

1. Verwenden der Trace-Funktion in der CES-Konsole

Mit der Trace-Funktion in der Cisco Email Security (CES)-Konsole können Administratoren simulieren und analysieren, wie eine bestimmte Nachricht verarbeitet wird. Dies ist eine der zuverlässigsten Methoden zur Validierung des Filterverhaltens vor der Durchsetzung.

Trace bietet Transparenz in:

- Analysieren der Nachricht
- Welche Filter werden evaluiert?
- Ob die Bedingung ausgelöst wird
- Die Reihenfolge der Regelausführung

Da die ESA MIME-Parsing, die Header-Normalisierung und die Inhaltsdekodierung durchführt, kann sich das Verhalten innerhalb der Appliance von dem externer Regex-Testtools unterscheiden. Detaillierte Anweisungen hierzu erhalten Administratoren in der offiziellen Cisco Dokumentation:

https://www.cisco.com/c/en/us/td/docs/security/ces/ces_16-0-3/user_guide/b_ESA_Admin_Guide_ces_16-0-3/b_ESA_Admin_Guide_12_1_chapter_0101001.html

Durch Verwendung von Trace wird sichergestellt, dass sich der Filter innerhalb der realen Verarbeitungsengine wie erwartet verhält.

2. Erstellen des Filters mit einer Protokollierungsaktion

Ein weiterer sicherer und empfohlener Ansatz besteht darin, den Filter mit einer unterbrechungsfreien Aktion wie Protokollierung bereitzustellen, anstatt eine aggressive Aktion wie das Verwerfen, Bouncing oder Quarantäne von Nachrichten anzuwenden.

Wenn der Filter so konfiguriert wird, dass er einen Eintrag protokolliert, wenn er zugeordnet wird, haben Administratoren folgende Möglichkeiten:

- Beobachtung der Übereinstimmungsfrequenz
- Erkennen unerwarteter Auslöser
- Validierung der Performance-Auswirkungen
- Analyse des tatsächlichen Datenverkehrsverhaltens

Mit diesem Ansatz wird der Filter effektiv in eine kontrollierte Überwachungsphase innerhalb des Produktionsdatenverkehrs versetzt. Sobald eine ausreichende Validierung abgeschlossen und das Verhalten als korrekt bestätigt wurde, kann die Aktion sicher in den Durchsetzungsmodus geändert werden.

Einführung des Ausdrucks in einem Content-Filter und in einem Wörterbuch

Sobald der reguläre Ausdruck ordnungsgemäß entworfen und validiert wurde, besteht der nächste Schritt darin, zu verstehen, wie er in die Cisco ESA eingegeben werden muss. Die Syntax kann leicht unterschiedlich aussehen, je nachdem, ob der Ausdruck direkt in einer Content-Filter-

Bedingung oder in einem Dictionary konfiguriert ist. Dieser Unterschied führt oft zu Verwirrung.

Einführung des Ausdrucks in einem Content-Filter

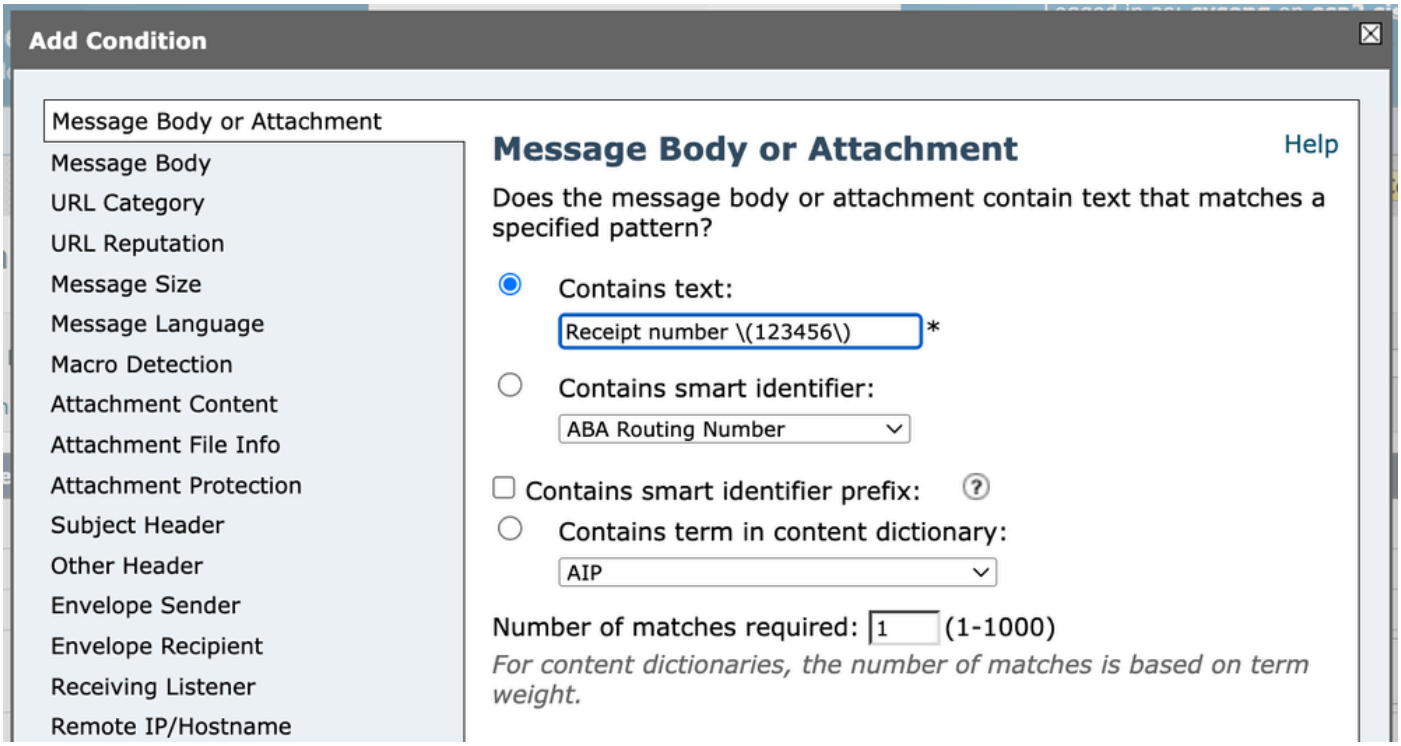
Beim Konfigurieren einer Content-Filter-Bedingung (z. B. durch Übereinstimmung mit dem Betreff-Header) muss der reguläre Ausdruck in das Bedingungsfeld eingegeben werden. Wenn der Text dem Literaltext entsprechen soll:

```
Receipt number (123456)
```

Wir müssen die Klammern umgehen, da sie Sonderzeichen in regulären Ausdrücken sind.

Daher muss der reguläre Ausdruck selbst wie folgt geschrieben werden:

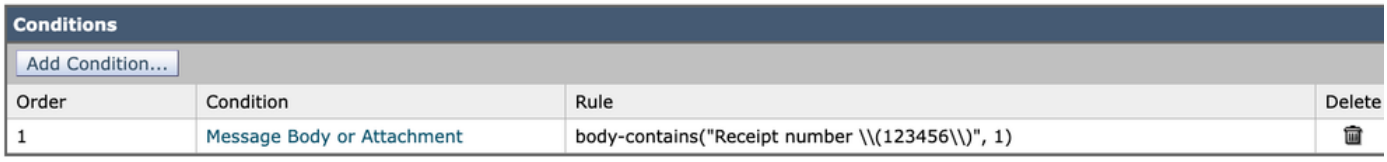
```
Receipt number \(123456\)
```



Content-Filter 1

Wenn Sie jedoch die vollständige Filterbedingung in der GUI oder der erweiterten Konfigurationsausgabe anzeigen, kann dies wie folgt aussehen:

```
subject == "Receipt number \\(123456\\)"
```



Content-Filter 2

Das kann auf den ersten Blick verwirrend sein. Der Grund für die doppelten umgekehrten Schrägstriche (\\) ist, dass der umgekehrte Schrägstrich selbst auch ein Sonderzeichen in zitierten Zeichenfolgen ist. In diesem Zusammenhang wird ein umgekehrter Schrägstrich verwendet, um die Klammer für die Regex-Engine zu umgehen, und der zweite umgekehrte Schrägstrich wird verwendet, um den umgekehrten Schrägstrich innerhalb der Zeichenfolge in Anführungszeichen zu umgehen.

Konkret:

\(123456\) ist der tatsächliche reguläre Ausdruck.

\\(zeigt an, wie das System \(in einem in Anführungszeichen gesetzten Konfigurationsstring darstellt.

Obwohl die Anzeige anders aussieht, bleibt der auszuwertende logische reguläre Ausdruck:

Belegnummer \(123456\)

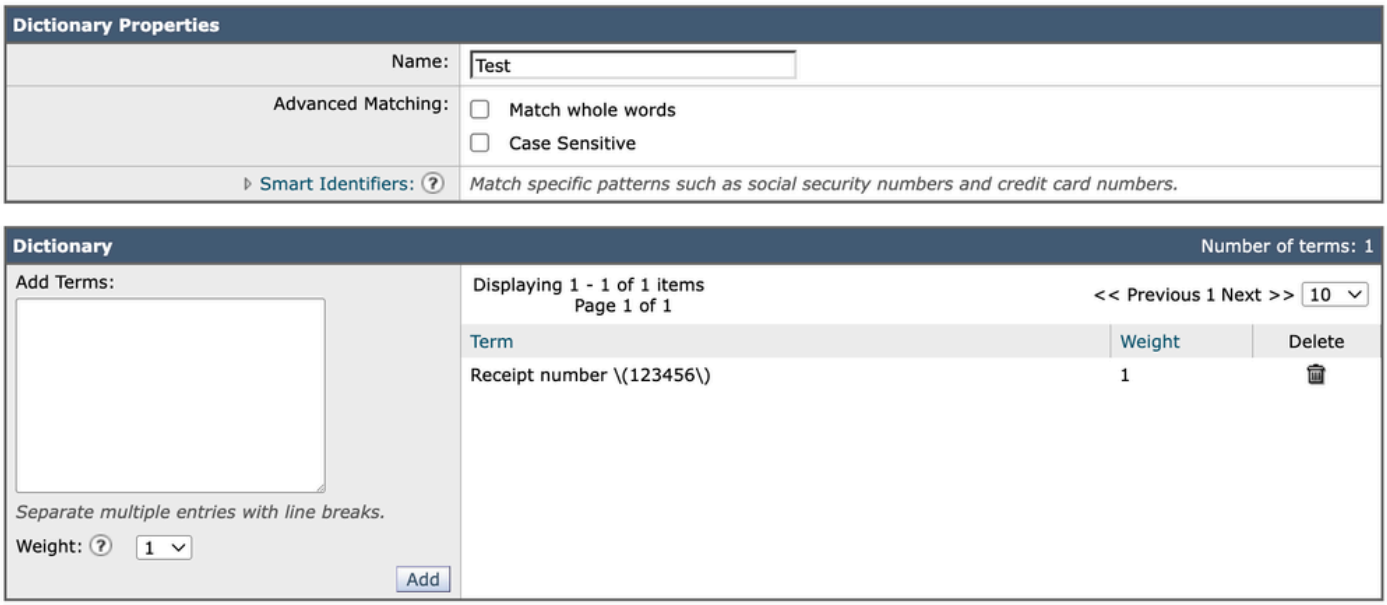
Es handelt sich hierbei lediglich um eine Zeichenfolge, die in der Konfigurationsausgabe entweicht.

Einführung in einen Ausdruck in einem Dictionary

Wenn Sie einem Dictionary denselben Ausdruck hinzufügen, wird der Eintrag wie folgt direkt eingefügt:

```
Receipt number \(123456\)
```

In diesem Fall wird es weiterhin genau wie geschrieben angezeigt. Im Gegensatz zur grafischen Benutzeroberfläche für den Inhaltsfilter sind für Wörterbücher keine zusätzlichen Ebenen mit Escapezeichen im visuellen Konfigurationsformat erforderlich.



Wörterbuch

Jeder Wörterbucheintrag wird je nach Struktur entweder als Klartext oder als regulärer Ausdruck ausgewertet. Wenn Sonderzeichen enthalten sind (in diesem Fall Klammern), muss der Ausdruck bereits bei der Eingabe ordnungsgemäß mit Escapezeichen versehen werden.

Über "Ganze Wörter zuordnen"

Bei der Konfiguration eines Wörterbuchs gibt es eine Option namens "Ganze Wörter zuordnen". In vielen Fällen wird empfohlen, sich beim Arbeiten mit regulären Ausdrücken nicht auf diese Einstellung zu verlassen.

Der Grund dafür ist, dass das Verhalten von Wortgrenzen mithilfe von Regex-Ankern genauer gesteuert werden kann.

Beispiele:

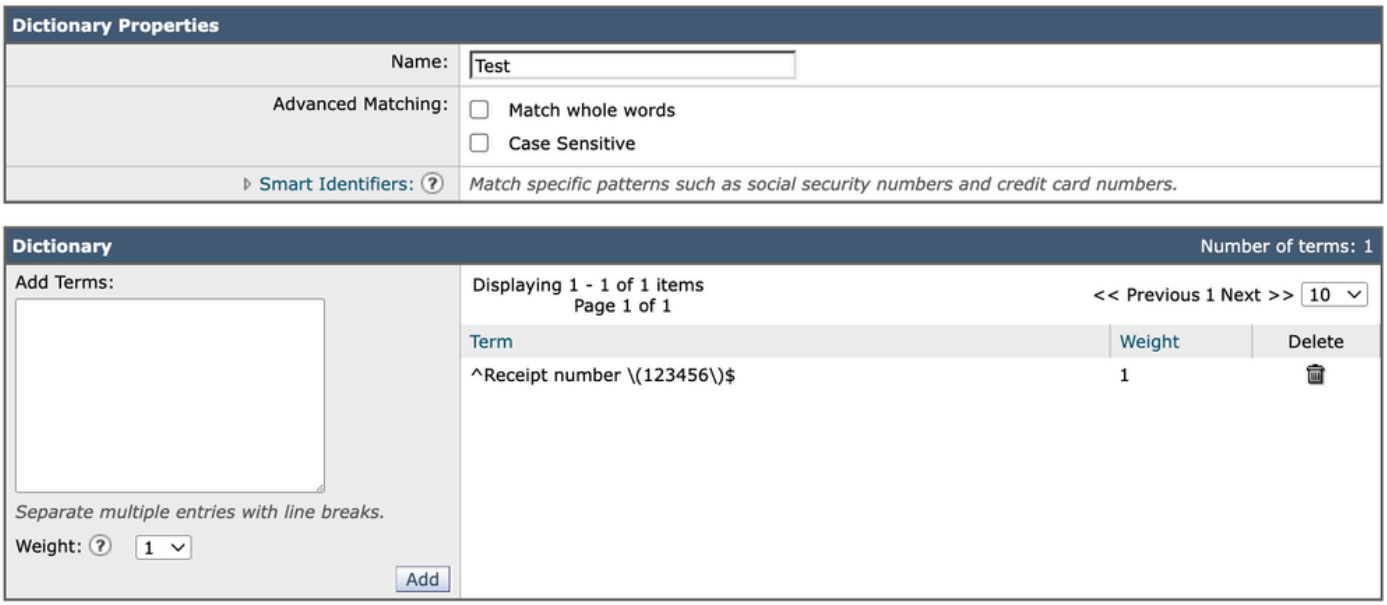
^ stellt sicher, dass die Übereinstimmung am Anfang beginnt.

$ stellt sicher, dass das Spiel am Ende endet.

Verwenden von Referenzzeichen wie:

```
^Receipt number \(123456\)$
```

Bietet explizite und vorhersehbare Kontrolle über das genaue Übereinstimmungsverhalten. Dieser Ansatz vermeidet potenzielle Unklarheiten im Zusammenhang mit der Interpretation von Wortgrenzen, insbesondere in mehrsprachigen oder nicht-ASCII-Umgebungen.



Wörterbuch 2

Aus diesem Grund ist es im Allgemeinen besser, die Genauigkeit der Übereinstimmung direkt im regulären Ausdruck zu verwalten, als sich auf die Option "Ganze Wörter zuordnen" zu verlassen.

Das Verständnis dieser feinen Unterschiede zwischen Content-Filtern und Wörterbüchern gewährleistet, dass sich Ausdrücke konsistent verhalten und reduziert das Risiko von Konfigurationsfehlern während der Implementierung.

Kosteneinstufung bei Regex auf Cisco ESA

Bei der Arbeit mit regulären Ausdrücken in Cisco ESA hängt die Auswirkung auf die Leistung in hohem Maße davon ab, wie viel Text die Engine scannen muss und wie viel Backtracking sie ausführen muss. Da die ESA ganze Nachrichtentexte, MIME-Teile und sogar decodierte Anhänge auswerten muss, können ineffiziente Muster die CPU-Auslastung deutlich erhöhen.

Es ist ein praktisches Ranking von den höchsten Berechnungskosten zu den niedrigsten.

Teuerste - Muster mit hohem Risiko

Diese Ausdrücke können sich erheblich auf die Leistung auswirken, insbesondere bei großen Nachrichten.

Verschachtelte Quantifizierer (Worst Case)

Beispiele:

```
(.*)+
(.+)+
(\S+)+
```

Diese sind extrem gefährlich, da sie exponentielle Backtracking-Szenarien erstellen.

Ein Quantifizierer in einem anderen Quantifizierer zwingt die Regex-Engine, viele Kombinationen auszuprobieren, bevor er ausfällt.

Bei echtem Datenverkehr kann dies zu erheblichen CPU-Spitzen führen.

Empfehlung: Vermeiden Sie unbegrenzte und mehrdeutige verschachtelte Quantifizierer.

Greedy .*, gefolgt von einem erforderlichen Muster

Beispiel:

```
.*text
.*\/\?text
```

Dieses Muster nimmt zunächst die gesamte Nachricht auf und verfolgt dann Zeichen für Zeichen zurück, bis es die erforderliche Teilzeichenfolge findet.

Wenn das Muster nicht vorhanden ist oder nahe dem Ende erscheint, verfolgt der Motor das erforderliche Token zurück und testet es an vielen Positionen, was die CPU-Kosten erhöht.

In der ESA, wo Körper groß sein können und MIME-Inhalt enthalten, wird dies sehr schnell teuer.

Empfehlung: Nicht vor .* setzen, um Teilzeichenfolgen zu erkennen. Die ESA sucht bereits nach den ausgewerteten Inhalten, und führende Platzhalter erhöhen nur die Rückverfolgung und CPU-Auslastung.

```
text$
\/\?text$
```

Große Alternativen mit freigegebenen Präfixen

Beispiel:

```
(a.*b|a.*c|a.*d)
```

Wenn mehrere Alternativen eine gemeinsame Struktur haben, wertet der Motor jeden Zweig nacheinander aus.

Wenn frühe Zweige fast übereinstimmen, aber spät ausfallen, versucht der Motor es erneut.

Dies erhöht die Auswertezeit erheblich.

Mittlere Kosten - Verwendung mit Vorsicht

Diese Muster sind nicht katastrophal, können aber immer noch ineffizient sein.

Breite .* Nutzung

Beispiel:

```
https://.*\?text
```

Obwohl nicht exponentiell, .* noch unbegrenzte Anpassung ermöglicht. Wenn die erwartete Teilzeichenfolge nicht schnell angezeigt wird, scannt das Modul große Teile der Nachricht.

In der ESA ist dies üblich, wenn E-Mail-Körper nach Phishing-URLs durchsucht werden.

Lazy Quantifiers (+?, *?)

Beispiel:

```
\S+?
.*?
```

Faule Quantifizierer ändern die Matching-Strategie (Shortest-First). Sie können Überübereinstimmungen in einigen Mustern reduzieren, aber bei großen "Such"-Workloads können sie die Anzahl der Versuche erhöhen, wenn das abschließende Token zu spät kommt oder fehlt.

In vielen ESA-Anwendungsfällen bieten sie keinen echten Nutzen und können unnötige interne Wiederholungsversuche einführen.

Sehr generische Zeichenklassen

Beispiele:

```
\S+
.+
```

Diese ermöglichen einen großen Spielbereich und erhöhen die Anzahl potenzieller Backtracking-

Pfade.

Speziellere Zeichenklassen sind immer vorzuziehen.

Niedrige Kosten - sichere und effiziente Muster

Diese werden für ESA-Produktionsumgebungen empfohlen.

Feste Literale

Beispiele:

```
text
iw\.adc
```

Literale Zeichenfolgen sind die effizientesten Übereinstimmungen. Der Motor führt einfache Vergleiche mit minimalem Overhead durch.

Verwenden von Referenzzeichen zum Einschränken des Suchbereichs

Wenn die Übereinstimmung an einer bestimmten Position erwartet wird, sollten Sie das Muster mit ^ oder $ verankern. Anker beschränken die Auswertung auf feste Positionen und verhindern, dass der Motor den gesamten Inhalt unnötig abtastet. Dies kann die Rückverfolgung reduzieren und die Leistung verbessern, insbesondere in großen Nachrichtenhauptteilen oder strukturierten Headern.

```
^Invoice$
```

Spezifische Zeichenklassen

```
[A-Za-z0-9.-]+
[^/\s]+
```

Diese schränken die Übereinstimmung ein, reduzieren den Suchraum erheblich und schränken die Rückverfolgung ein.

Strukturierte und eingeschränkte Muster

Beispiel:

```
https?:\/\/[A-Za-z0-9.-]+(?:\/[^?\s]*)*\/\?text
```

- Die Domäne ist repariert.
- Keine Verwendung von .*.
- enthält keine katastrophalen verschachtelten Muster (Beispiel: (.*)+)
- Keine unnötigen faulen Bediener.
- Jeder Abschnitt ist eingeschränkt.

Dadurch werden die Auswirkungen auf die CPU im Vergleich zum breit gefächerten Platzhalterabgleich erheblich reduziert.

Praktische Anleitung für die Cisco ESA

Beim Entwerfen von regulären Ausdrücken für Nachrichten- oder Inhaltsfilter:

1. Je genauer das Muster ist, desto besser ist die Leistung.
2. Vermeiden Sie .*, es sei denn, es ist wirklich notwendig — und vermeiden Sie insbesondere, erforderliche Token danach zu platzieren.
3. Verwenden Sie niemals geschachtelte Quantifizierer.
4. Bevorzugen explizite Zeichenklassen gegenüber Platzhaltern.
5. Testen Sie neue Ausdrücke immer im Überwachungsmodus, bevor Sie sie erzwingen.

Regex-Leistungsvergleich (Cisco ESA-Kontext)

Muster	Empfohlen	Risiko zurückverfolgen	ESA-Einfluss	Empfohlene Alternative
https?:\/\/.*/\?text.*	Nein	Hoch	Höher	^https?:\/\/[A-Za-z0-9.-]+(?:\/[^?\s]*)*\/\?text
https?:\/\/.*\?text	⚠ Mit Vorsicht	Mittel-Hoch	Mittel-Hoch	^https?:\/\/[^?\s]+\?text$
https?:\/\/.*	Nein	Mittel-Hoch	Mittel	^https?:\/\/[A-Za-z0-9.-]+(
.*Kennwort	Nein	Hoch	Höher	Kennwort$
.*Text.*	Nein	Hoch	Höher	Text
.*(Rechnung|Zahlung|Überweisung)	Nein	Hoch	Höher	(Rechnung|Zahlung|Über

(.+)+	Nie	Sehr hoch (exponentiell)	Schwerwiegend	Restrukturierung ohne verschachtelte Quantifizi (Beispiel .+)
.*@.*	Nein	Hoch	Höher	[A-Za-z0-9._%+-]+@[A-Z]+\.[A-Za-z]{2,}
\S+?	Nicht ideal	Mittel	Mittel	\S+ oder spezifischere K [A-Za-z0-9.-]+
.*\/admin	Nein	Hoch	Höher	\/Administrator$
.*(Anmeldung|Überprüfung).*	Nein	Hoch	Höher	(Anmeldung|bestätigen)
^.*Text	Nein	Hoch	Höher	text$ (oder ^text, wenn d wichtig ist)

Schlussfolgerung

Reguläre Ausdrücke sind ein leistungsstarkes und flexibles Tool innerhalb der Cisco ESA, das eine präzise Inhaltsüberprüfung und eine erweiterte Richtliniendurchsetzung sowohl bei Nachrichtenfiltern als auch bei Inhaltsfiltern ermöglicht. Mit dieser Flexibilität geht jedoch auch Verantwortung einher. Unzureichend entwickelte oder nicht ausreichend getestete Ausdrücke können zu Fehlalarmen, nicht erkannten Bedrohungen, Leistungseinbußen oder einer unbeabsichtigten Unterbrechung des legitimen E-Mail-Verkehrs führen.

Aus diesem Grund muss die Verwendung regulärer Ausdrücke in der ESA immer strukturiert und diszipliniert sein. Die Erstellungsphase muss sicherstellen, dass der Ausdruck syntaktisch korrekt, ordnungsgemäß entzweit, effizient und logisch auf das beabsichtigte Ziel ausgerichtet ist. Externe Tools und KI-gestützte Generierung können diesen Prozess erheblich beschleunigen, sie dürfen jedoch niemals eine sorgfältige Validierung ersetzen.

Ebenso wichtig ist die Validierungsphase innerhalb der ESA-Umgebung selbst. Da die ESA Nachrichten durch MIME-Parsing, Header-Normalisierung und Content-Decodierung verarbeitet, kann das Verhalten in der realen Welt von den theoretischen Erwartungen abweichen. Durch die Verwendung von Tools wie Trace und die anfängliche Bereitstellung von Filtern im Protokollierungs- oder Überwachungsmodus können Administratoren das korrekte Verhalten ohne Betriebsrisiko bestätigen.

Zusammenfassend lässt sich sagen, dass reguläre Ausdrücke so einfach wie möglich gehalten, gründlich getestet und mit Vorsicht bereitgestellt werden müssen. Ein gut durchdachter und validierter Filter sorgt nicht nur für eine effektive Richtliniendurchsetzung, sondern schützt auch

die Systemstabilität und gewährleistet ein vorhersehbares Verhalten in Produktionsumgebungen.

Dokumentation

Weitere technische Details und offizielle Anleitungen zur Implementierung und Verwendung regulärer Ausdrücke innerhalb der Cisco ESA erhalten Administratoren in der Cisco Produktdokumentation.

Der Abschnitt "Reguläre Ausdrücke in Regeln" bietet einen Überblick darüber, wie reguläre Ausdrücke innerhalb von Nachrichtenfiltern und Inhaltsfiltern ausgewertet werden, einschließlich Syntaxüberlegungen und der Verwendung innerhalb von Regelbedingungen.

https://www.cisco.com/c/en/us/td/docs/security/ces/ces_16-0-3/user_guide/b_ESA_Admin_Guide_ces_16-0-3/b_ESA_Admin_Guide_12_1_chapter_01000.html#con_1192755

Der Abschnitt "Richtlinien für die Verwendung regulärer Ausdrücke" enthält praktische Empfehlungen zur korrekten Syntax, zur Verankerung von Ausdrücken, zum Umgang mit Sonderzeichen und zur Vermeidung gängiger Fehler, die die Leistung oder die Genauigkeit der Übereinstimmung beeinträchtigen können.

https://www.cisco.com/c/en/us/td/docs/security/ces/ces_16-0-3/user_guide/b_ESA_Admin_Guide_ces_16-0-3/b_ESA_Admin_Guide_12_1_chapter_01000.html#con_1125696

Die Überprüfung dieser offiziellen Ressourcen wird dringend empfohlen, wenn Sie Filter entwerfen oder eine Fehlerbehebung durchführen, die auf regulären Ausdrücken basieren, da sie eine zuverlässige Anleitung bieten, die mit der jeweiligen verwendeten AsyncOS-Version übereinstimmt.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.