Règles de prévention des intrusions personnalisées

Les rubriques suivantes décrivent comment utiliser l'éditeur de règles de prévention des intrusions :

Présentation des règles de prévention des intrusions personnalisées

Une *règle de prévention des intrusions* est un ensemble de mots-clés et d'arguments que le système utilise pour détecter les tentatives d'exploitation des vulnérabilités de votre réseau. Lorsque le système analyse le trafic réseau, il compare les paquets en fonction des conditions spécifiées dans chaque règle. Si les données du paquet correspondent à toutes les conditions spécifiées dans une règle, la règle se déclenche. Si une règle est une *règle d'alerte*, un incident d'intrusion est généré. S'il s'agit d'une *règle de réussite*, le trafic est ignoré. Pour une règle de *suppression* lors d'un déploiement en ligne, le système abandonne le paquet et génère un événement. Vous pouvez afficher et évaluer les incidents d'intrusion à partir de l'interface Web Cisco Secure Firewall Management Center.

Le système Firepower fournit deux types de règles de prévention des intrusions : des règles d'objet partagé et des règles de texte standard. Les Talos Intelligence Group peuvent utiliser des règles d'objet partagé pour détecter les attaques contre les vulnérabilités contrairement aux règles de texte standard traditionnelles. Vous ne pouvez pas créer de règles d'objet partagé. Lorsque vous écrivez votre propre règle de prévention des intrusions, vous créez une règle de texte standard.

Vous pouvez rédiger des règles de texte standard personnalisées pour ajuster les types d'événements que vous êtes susceptible de voir. Notez que même si cette documentation aborde parfois les règles visant à détecter des exploits spécifiques, les règles les plus efficaces ciblent le trafic qui peut tenter d'exploiter des vulnérabilités connues plutôt que des exploits connus spécifiques. En écrivant des règles et en spécifiant le message

d'événement de la règle, vous pouvez plus facilement identifier le trafic qui indique des attaques et des contournements de politiques.

Lorsque vous activez une règle de texte standard personnalisée dans une politique de prévention des intrusions personnalisée, gardez à l'esprit que certains mots-clés et certains arguments de règle nécessitent que le trafic soit d'abord décodé ou prétraité d'une certaine manière. Ce chapitre explique les options que vous devez configurer dans votre politique d'analyse de réseau, qui régit le prétraitement. Notez que si vous désactivez un préprocesseur requis, le système l'utilise automatiquement avec ses paramètres actuels, bien que le préprocesseur reste désactivé dans l'interface Web de politique d'analyse de réseau.

Mise en garde Veillez à utiliser un environnement réseau contrôlé pour tester les règles de prévention des intrusions que vous écrivez avant de les utiliser dans un environnement de production. Des règles de prévention des intrusions mal écrites peuvent sérieusement affecter les performances du système.

Dans un déploiement multidomaine, le système affiche les règles créées dans le domaine actuel, que vous pouvez modifier. Il affiche également les règles créées dans les domaines ancêtres, que vous ne pouvez pas modifier. Pour afficher et modifier les règles créées dans un domaine inférieur, basculez vers ce domaine. Les règles de prévention des intrusions fournies par le système appartiennent au domaine global. Les administrateurs des domaines descendants peuvent créer des copies modifiables localement de ces règles système.

Exigences de licence pour l'éditeur de règles de prévention des intrusions

Licence de défense contre les menaces

IPS

Licence traditionnelle

Protection

Exigences et conditions préalables de l'éditeur de règles de prévention des intrusions

Prise en charge des modèles

Tout.

Domaines pris en charge

N'importe quel

Rôles utilisateur

- Admin

- Administrateur d’intrusion

Anatomie des règles

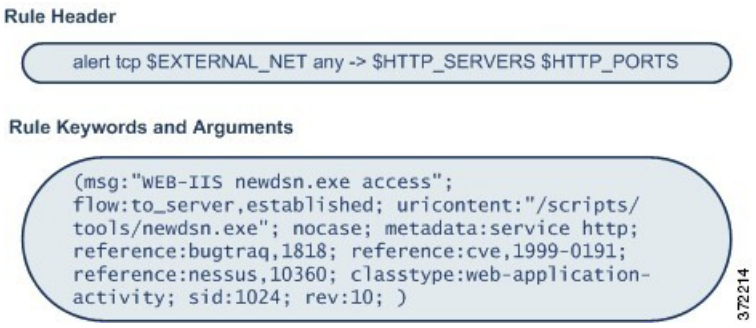
Toutes les règles de texte standard contiennent deux sections logiques : l’en-tête de règle et les options de règle. L’en-tête de règle contient :

- l’action ou le type de règle
- le protocole
- les adresses IP et les masques de réseau de la source et de la destination
- des indicateurs de direction indiquant le flux du trafic de la source à la destination
- les ports de source et de destination

La section des options de règle contient :

- les messages d'événements
- les mots-clés, leurs paramètres et leurs arguments.
- les schémas auxquels la charge utile d’un paquet doit correspondre pour déclencher la règle
- les spécifications des parties du paquet que le moteur de règles doit inspecter

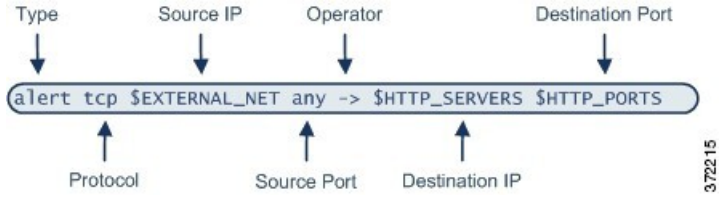
Le diagramme suivant illustre les parties d’une règle :



Notez que la section des options d’une règle est la section entre parenthèses. L’éditeur de règles de prévention des intrusions fournit une interface facile à utiliser pour vous aider à créer des règles de texte standard.

En-tête de règle de prévention des intrusions

Chaque règle de texte standard et règle d’objet partagé possède un en-tête de règle contenant des paramètres et des arguments. La figure suivante illustre les parties d’un en-tête de règle :



Le tableau suivant décrit chaque partie de l'en-tête de règle ci-dessus.

Tableau 1 : Valeurs d'en-tête de règle

Composant d'en-tête de règle	Exemple de valeur	Cette valeur...
Action	`alerte`	Génère un incident d'intrusion lorsqu'elle est déclenchée.
Protocole	`tcp`	Teste le trafic TCP uniquement.
Source IP Address (adresse IP source)	`$EXTERNAL_NET`	Teste le trafic provenant de tout hôte qui ne fait pas partie de votre réseau interne.
Ports sources	`Tous`	Teste le trafic provenant de n'importe quel port de l'hôte d'origine.
Opérateur	`->`	Teste le trafic externe (destiné aux serveurs Web de votre réseau).
Destination IP Address (adresse IP de destination)	`$HTTP_SERVERS`	Teste le trafic à acheminer vers n'importe quel hôte défini comme serveur Web sur votre réseau interne.
Ports de destination	`$HTTP_PORTS`	Teste le trafic acheminé vers un port HTTP sur votre réseau interne.

Remarque L'exemple précédent utilise des variables par défaut, comme la plupart des règles de prévention des intrusions.

Sujets connexes

Ensemble de variables

Action d'en-tête de règle de prévention des intrusions

Chaque en-tête de règle comprend un paramètre qui spécifie l'action que le système exécute lorsqu'un paquet déclenche une règle. Les règles avec l'action définie sur *alert* génèrent un incident d'intrusion pour le paquet qui a déclenché la règle et enregistrent les détails de ce paquet. Les règles avec l'action définie avec *pass* ne génèrent pas d'événement pour le paquet qui a déclenché la règle et n'enregistrent pas les détails dudit paquet.

Remarque Dans un déploiement en ligne, les règles dont l'état est *Abandonner et Générer des événements* génèrent un incident d'intrusion pour le paquet qui a déclenché la règle. En outre, si vous appliquez une règle de suppression dans un déploiement passif, la règle sert de règle d'alerte.

Par défaut, les règles de réussite remplacent les règles d'alerte. Vous pouvez créer des règles de réussite pour empêcher les paquets qui répondent aux critères définis dans la règle de réussite de déclencher l'application de la règle d'alerte dans des situations spécifiques, plutôt que de désactiver la règle d'alerte. Par exemple, vous pouvez souhaiter qu'une règle qui recherche les tentatives de connexion à un serveur FTP en tant qu'utilisateur « anonyme » reste active. Cependant, si votre réseau comporte un ou plusieurs serveurs FTP anonymes légitimes, vous pouvez écrire et activer une règle de réussite qui spécifie que, pour ces serveurs spécifiques, les utilisateurs anonymes ne déclenchent pas la règle d'origine.

Dans l'éditeur de règles de prévention des intrusions, sélectionnez le type de règle dans la liste **Action**.

Protocole d'en-tête de règle de prévention des intrusions

Dans chaque en-tête de règle, vous devez préciser le protocole du trafic inspecté par la règle. Vous pouvez spécifier les protocoles de réseau suivants pour l'analyse :

- Internet Control Message Protocol (protocole ICMP)
- IP (protocole Internet)

Remarque Le système ignore les définitions de port dans un en-tête de règle de prévention des intrusions lorsque le protocole est défini sur `ip`.

- protocole TCP (Transmission Control Protocol)
- User Datagram Protocol (protocole UDP)

Utilisez **IP** comme type de protocole pour examiner tous les protocoles attribués par l'IANA, y compris TCP, UDP, ICMP, IGMP et bien d'autres.

Remarque Vous ne pouvez actuellement pas écrire de règles qui correspondent aux modèles de l'en-tête suivant (par exemple, l'en-tête TCP) d'une charge utile IP. Au lieu de cela, les correspondances de contenu commencent par le dernier protocole décodé. Comme solution de contournement, vous pouvez mettre en correspondance des schémas dans les en-têtes TCP en utilisant les options de règles.

Dans l'éditeur de règles de prévention des intrusions, vous sélectionnez le type de protocole dans la liste **Protocol**.

Sujets connexes

Protocole d'en-tête de règle de prévention des intrusions, à la page 5

Direction de l'en-tête de la règle de prévention des intrusions

Dans l'en-tête de règle, vous pouvez préciser la direction dans laquelle le paquet doit se déplacer pour que la règle puisse l'inspecter. Le tableau suivant décrit ces options.

Tableau 2 : Options directionnelles des en-têtes de règles

Utiliser...	Pour tester...
Directionnel	uniquement le trafic de l'adresse IP source spécifiée vers l'adresse IP de destination spécifiée
Bidirectionnel	tout le trafic circulant entre les adresses IP source et de destination précisées

Adresses IP de source et de destination de l'en-tête de règle de prévention des intrusions

Restreindre l'inspection de paquets aux paquets provenant d'adresses IP spécifiques ou destinés à une adresse IP spécifique réduit la quantité d'inspection de paquets que le système doit effectuer. Cela réduit également les faux positifs en rendant la règle plus spécifique et en éliminant la possibilité que la règle se déclenche pour les paquets dont les adresses IP de source et de destination n'indiquent pas un comportement suspect.

Astuces Le système reconnaît uniquement les adresses IP et n'accepte pas les noms d'hôte pour les adresses IP source ou de destination.

Dans l'éditeur de règles de prévention des intrusions, vous spécifiez les adresses IP de source et de destination dans les champs **Source IPs** et **Destination IPs**.

Lors de la rédaction de règles de texte standard, vous pouvez spécifier les adresses IPv4 et IPv6 de différentes manières, selon vos besoins. Vous pouvez spécifier une adresse IP unique, `n'importe quelle`, des listes d'adresses IP, une notation CIDR, des longueurs de préfixes ou une variable de réseau. En outre, vous pouvez indiquer que vous souhaitez exclure une adresse IP spécifique ou un ensemble d'adresses IP. Lorsque vous spécifiez des adresses IPv6, vous pouvez utiliser n'importe quelle convention d'adressage définie dans la RFC 4291.

Syntaxe de l'adresse IP dans les règles de prévention des intrusions

Le tableau suivant résume les différentes façons dont vous pouvez spécifier des adresses IP source et de destination.

Tableau 3 : Syntaxe de l'adresse IP source/destination

Pour indiquer...	Utiliser...	Exemple
toute adresse IP	`Tous`	`Tous`
une adresse IP précise	l'adresse IP Notez que vous ne devez pas combiner les adresses source et de destination IPv4 et IPv6 dans une même règle.	`192.168.1.1` `2001:db8::abcd`
une liste d'adresses IP	Des crochets (`[]`) pour délimiter les adresses IP et des virgules pour les séparer	`[192.168.1.1,192.168.1.15]` `[2001:db8::b3ff, 2001:db8::0202]`
un bloc d'adresses IP	Bloc CIDR IPv4 ou notation des préfixes d'adresses IPv6	`192.168.1.0/24` `2001:db8::/32`

Pour indiquer...	Utiliser...	Exemple
tout sauf une adresse IP spécifique ou un ensemble d'adresses	le caractère `!` avant l'adresse ou les adresses IP que vous souhaitez annuler	`!192.168.1.15` `!2001:db8::0202:b3ff:fe1e`
tout ce qui se trouve dans un bloc d'adresses IP, à l'exception d'une ou de plusieurs adresses IP spécifiques	un bloc d'adresses suivi d'une liste d'adresses ou de blocs annulés	`[10.0.0/8, !10.2.3.4, !10.1.0.0/16]` `[2001:db8::/32, !2001:db8::8329, !2001:db8::0202]`
Adresses IP définies par une variable de réseau	le nom de la variable, en lettres majuscules, précédé de `$` Notez que les règles de préprocesseur peuvent déclencher des événements quels que soient les hôtes définis par les variables de réseau utilisées dans les règles de prévention des intrusions.	`$HOME_NET`
toutes les adresses IP, à l'exception des adresses définies par une variable d'adresse IP	le nom de la variable, en lettres majuscules, précédé de `!$`	`!$HOME_NET`

Les descriptions suivantes fournissent des renseignements supplémentaires sur certaines des méthodes de saisie de l'adresse IP.

toute adresse IP

Vous pouvez définir le mot « `any` » comme adresse IP de source ou de destination pour indiquer une adresse IPv4 ou IPv6.

Par exemple, la règle suivante utilise l'argument **`any`** dans les champs **IP source** et **IP de destination** et évalue les paquets avec toute adresse de source ou de destination IPv4 ou IPv6 :

```
alert tcp any any -> any any
```

Vous pouvez également utiliser `::` pour indiquer n'importe quelle adresse IPv6.

Adresses IP multiples

Vous pouvez répertorier les adresses IP individuelles en les séparant par des virgules et, éventuellement, en entourant les listes ne faisant pas l'objet de négation de parenthèses, comme le montre l'exemple suivant :

```
[192.168.1.100,192.168.1.103,192.168.1.105]
```

Vous pouvez répertorier les adresses IPv4 et IPv6 seules ou dans n'importe quelle combinaison, comme le montre l'exemple suivant :

```
[192.168.1.100,2001:db8::1234,192.168.1.105]
```

Notez qu'entourer une liste d'adresses IP de parenthèses, qui était obligatoire dans les versions antérieures du logiciel, n'est pas obligatoire. Notez également que vous pouvez éventuellement saisir des listes avec un espace avant ou après chaque virgule.

Remarque Vous devez entourer les listes annulées de parenthèses.

Vous pouvez également utiliser la notation CIDR (Classless Inter-Domain Routing) IPv4 ou les longueurs de préfixe IPv6 pour spécifier les blocs d'adresses. Par exemple :

- 192.168.1.0/24 spécifie les adresses IPv4 dans le réseau 192.168.1.0 avec un masque de sous-réseau de 255.255.255.0, c'est-à-dire de 192.168.1.0 à 192.168.1.255.
- 2001:db8::/32 précise les adresses IPv6 dans le réseau 2001:db8:: avec une longueur de préfixe de 32 bits, c'est-à-dire 2001:db8:: à 2001:db8:ffff:ffff:ffff:ffff:ffff: ffff.

Astuces Si vous devez spécifier un bloc d'adresses IP, mais que vous ne pouvez pas l'exprimer à l'aide de la notation CIDR ou de longueur de préfixe, vous pouvez utiliser des blocs CIDR et des longueurs de préfixe dans une liste d'adresses IP.

Négation des adresses IP

Vous pouvez utiliser un point d'exclamation (`!`) pour annuler une adresse IP précise. C'est-à-dire que vous pouvez mettre en correspondance n'importe quelle adresse IP à l'exception de l'adresse ou des adresses IP précisées. Par exemple, `!192.168.1.1` spécifie toute adresse IP autre que 192.168.1.1 et `!001:db8:ca2e::fa4c` spécifie toute adresse IP autre que 2001:db8:ca2e::fa4c.

Pour annuler une liste d'adresses IP, placez `!` avant une liste d'adresses IP entre parenthèses. Par exemple, `![192.168.1.1,192.168.1.5]` définirait toute adresse IP autre que 192.168.1.1 ou 192.168.1.5.

Remarque Vous devez utiliser des crochets pour annuler une liste d'adresses IP.

Soyez prudent lorsque vous utilisez le caractère de négation avec des listes d'adresses IP. Par exemple, si vous utilisez `[!192.168.1.1,!192.168.1.5]` pour mettre en correspondance toute adresse qui n'est pas 192.168.1.1 ou 192.168.1.5, le système interprète cette syntaxe comme « tout ce qui n'est pas 192.168.1.1, **ou** tout ce qui n'est pas 192.168.1.5. »

Comme 192.168.1.5 n'est pas 192.168.1.1 et que 192.168.1.1 n'est pas 192.168.1.5, les deux adresses IP correspondent à la valeur d'adresse IP de `[!192.168.1.1,!192.168.1.5]`, et c'est essentiellement la même chose que d'utiliser «`n'importe quel`.

Au lieu de cela, utilisez `![192.168.1.1,192.168.1.5]`. C'est-à-dire que le système interprète cela comme « **not** 192.168.1.1 **and not** 192.168.1.5 », ce qui correspond à toute adresse IP autre que celles indiquées entre parenthèses.

Notez que vous ne pouvez logiquement pas utiliser la négation avec `une` option qui, si elle était refusée, n'indiquerait aucune adresse.

Sujets connexes

Ensemble de variables

Ports source et de destination de l'en-tête de la règle de prévention des intrusions

Dans l'éditeur de règles de prévention des intrusions, vous spécifiez les ports source et de destination dans les champs **Source Port** (Port source) et **Destination Port** (Port de destination).

Syntaxe du port dans les règles de prévention des intrusions

Le système Firepower utilise un type de syntaxe spécifique pour définir les numéros de port utilisés dans les en-têtes de règles.

Remarque Le système ignore les définitions de port dans un en-tête de règle de prévention des intrusions lorsque le protocole est défini sur `ip`.

Vous pouvez répertorier les ports en les séparant par des virgules, comme le montre l'exemple suivant :

```
80, 8080, 8138, 8600-9000, !8650-8675
```

L'exemple suivant montre comment entourer une liste de ports entre parenthèses, ce qui était obligatoire dans les versions précédentes du logiciel, mais ne l'est plus :

```
[80, 8080, 8138, 8600-9000, !8650-8675]
```

Notez que vous **devez** entourer les listes de ports annulées entre parenthèses, comme le montre l'exemple suivant :

```
![20, 22, 23]
```

Le tableau suivant résume la syntaxe que vous pouvez utiliser :

Tableau 4 : Syntaxe du port source/destination

Pour indiquer...	Utiliser	Exemple
n'importe quel port	`Tous`	`Tous`
un port précis	le numéro de port	`80`
une plage de ports	un tiret entre le premier et le dernier numéro de port de la plage	`80-443`
tous les ports sont inférieurs ou égaux à un port spécifique	un tiret avant le numéro de port	`-21`
tous les ports sont supérieurs ou égaux à un port spécifique	un tiret après le numéro de port	`80-`
tous les ports, à l'exception d'un port ou d'une plage de ports en particulier	le `!` avant le port, la liste de ports ou la plage de ports que vous souhaitez exclure Notez que vous pouvez logiquement utiliser l'exclusion avec toutes les désignations de port, à l'exception de celle qui, si elle était refusée, indiquerait l'*absence de port.*	`!20`

Pour indiquer...	Utiliser	Exemple
tous les ports définis par une variable de port	le nom de la variable, en lettres majuscules, précédé de `$`	`$HTTP_PORTS`
tous les ports, à l'exception des ports définis par une variable de port	le nom de la variable, en lettres majuscules, précédé de `!$`	`!$HTTP_PORTS`

Détails des Événements liés aux intrusions

Lorsque vous élaborez une règle de texte standard, vous pouvez inclure des informations contextuelles qui décrivent la vulnérabilité que la règle détecte dans les tentatives d'exploitation. Vous pouvez également inclure des références externes aux bases de données de vulnérabilités et définir la priorité de l'événement dans votre organisation. Lorsque les analystes observent l'événement, ils disposent alors d'informations sur la priorité, l'exploitation et les mesures d'atténuation connues.

Message

Vous pouvez spécifier du texte significatif qui s'affiche comme message lorsque la règle se déclenche. Le message donne un aperçu immédiat de la nature de la vulnérabilité que la règle détecte les tentatives d'exploitation. Vous pouvez utiliser n'importe quel caractère ASCII standard imprimé, à l'exception des accolades (`{}`). Le système supprime les guillemets qui entourent complètement le message.

Astuces Vous devez spécifier un message de règle. En outre, le message ne peut pas contenir d'espaces blancs, d'un ou de plusieurs guillemets seulement, d'une ou de plusieurs apostrophes seulement ou d'une combinaison d'espaces, de guillemets ou d'apostrophes.

Pour définir le message d'événement dans l'éditeur de règles de prévention des intrusions, saisissez le message d'événement dans le champ **Message**.

Classification

Pour chaque règle, vous pouvez spécifier une classification d'attaque qui apparaît dans l'affichage de paquets de l'événement. Le tableau suivant dresse la liste du nom et du numéro pour chaque classification.

Tableau 5 : Classification des règles

Nombre	Nom de la classification	Description
1	not-suspicious	Trafic non suspect
2	inconnu	Trafic inconnu
3	bad-unknown	Trafic potentiellement néfaste
4	attempted-recon	Tentative de fuite d'informations
5	successful-recon-limited	Fuite d'informations
6	successful-recon-largescale	Fuite d'informations à grande échelle

Nombre	Nom de la classification	Description
7	attempted-dos	Tentative de dénis de service
8	successful-dos	Déni de service
9	attempted-user	Tentative d'obtention de privilèges d’utilisateur
10	unsuccessful-user	Échec de l’obtention de privilèges d’utilisateur
11	successful-user	Obtention de privilège d’utilisateur réussie
12	attempted-admin	Tentative d'obtention de privilèges d’administrateur
13	successful-admin	Obtention de privilège d’administrateur réussie
14	rpc-portmap-decode	Décodage d’une requête RPC
15	shellcode-detect	Du code d’exécutable a été détecté
16	string-detect	Une chaîne suspecte a été détectée
17	suspicious-filename-detect	Un nom de fichier suspect a été détecté
18	suspicious-login	Une tentative de connexion avec un nom d’utilisateur suspect a été détectée
19	system-call-detect	Un appel système a été détecté
20	tcp-connection	Une connexion TCP a été détectée
21	trojan-activity	Un cheval de Troie réseau a été détecté
22	unusual-client-port-connection	Un client utilisait un port inhabituel
23	analyse du réseau	Détection d’une analyse du réseau
24	denial-of-service	Détection d’une attaque par déni de service
25	non-standard-protocol	Détection d’un protocole ou d’un événement non standard
26	protocol-command-decode	Décodage de commande de protocole générique
27	web-application-activity	Accès à une application Web potentiellement vulnérable
28	web-application-attack	Attaque d'application Web
29	misc-activity	Activité diverse
30	misc-attack	Attaques diverses
31	icmp-event	Événement ICMP générique
32	inappropriate-content	Contenu inapproprié détecté

Nombre	Nom de la classification	Description
33	policy-violation	Violation potentielle de la politique d'entreprise
34	default-login-attempt	Tentative de connexion avec un nom d'utilisateur et un mot de passe par défaut
35	sdf	Données sensibles
36	malware-cnc	Trafic de commande et de contrôle de programmes malveillants connus
37	exploitation-côté client	Tentative d'exploitation connue du côté client
38	file-format	Fichier malveillant connu ou exploit basé sur un fichier

Classification personnalisée

Si vous souhaitez un contenu plus personnalisé pour la description de l'affichage de paquet des événements générés par une règle que vous définissez, vous pouvez créer une classification personnalisée.

Argument	Description
Nom de la classification	Le nom de la classification. La page est difficile à lire si vous utilisez plus de 40 caractères. Les caractères suivants ne sont pas pris en charge : `<>()\'"&$;` ainsi que le caractère espace.
Description de la classification	Une description de la classification. Vous pouvez utiliser des caractères alphanumériques et des espaces. Les caractères suivants ne sont pas pris en charge : `<>()\'"&$;`
Priorité	high (élevé), medium (moyen), or low (bas).

Priorité personnalisée

Par défaut, la priorité d'une règle découle de la classification d'événement pour la règle. Cependant, vous pouvez remplacer la priorité de classification d'une règle en ajoutant le mot-clé `priority` à la règle et en sélectionnant une priorité élevée, moyenne ou faible. Par exemple, pour attribuer une priorité élevée à une règle qui détecte les attaques d'applications Web, ajoutez le mot-clé `priority` à la règle et sélectionnez **high** comme priorité.

Référence personnalisée

Vous pouvez utiliser le mot-clé `reference` pour ajouter des références à des sites Web externes et des informations supplémentaires sur l'événement. L'ajout d'une référence fournit aux analystes une ressource immédiatement disponible pour les aider à déterminer pourquoi le paquet a déclenché une règle. Le tableau suivant répertorie certains des systèmes externes qui peuvent fournir des données sur les exploits et les attaques connus.

Tableau 6 : Systèmes d'identification des attaques externes

ID de système	Description	Exemple d'ID
`bugtraq`	Page Bugtraq	`8550`
`cve`	ID de vulnérabilités et risques courants	`2020-9607`
`mcafee`	Page McAfee	`98574`
`url`	Site Web de référence	`www.example.com?exploit=14`
`msb`	Bulletin de sécurité de Microsoft	`MS11-082`
`nessus`	Page Nessus	`10039`
`secure-url`	Référence de site Web sécurisé (https: //...)	`intranet/exploits/exploit=14` Notez que vous pouvez utiliser `secure-url` avec n'importe quel site Web sécurisé.

Vous spécifiez une référence en saisissant une valeur de référence, comme suit :

```
id_system,id
```

où id_system est le système utilisé comme préfixe et id est le numéro d'ID CVE, l'ID d'Arachnides ou l'URL (sans `http://`).

Par exemple, pour préciser le problème d'Adobe Acrobat et de Reader documenté dans CVE-2020-9607, saisissez la valeur :

```
cve,2020-9607
```

Tenez compte des éléments suivants lors de l'ajout de références à une règle :

- N'utilisez pas d'espace après la virgule.
- N'utilisez pas de lettres majuscules dans l'ID système.

Sujets connexes

Ajouter une classification personnalisée, à la page 13
Définition d'une priorité d'événement, à la page 14
Définition d'une référence d'événement, à la page 14

Ajouter une classification personnalisée

Dans un déploiement multidomaine, le système affiche les classifications personnalisées créées dans le domaine actuel, et vous pouvez définir les priorités de ces classifications. Il affiche également les classifications personnalisées créées dans les domaines ascendants, mais vous ne pouvez pas définir les priorités de ces classifications. Pour afficher et modifier les règles créées dans un domaine inférieur, basculez vers ce domaine.

Procédure

Étape 1 Lors de la création ou de la modification d'une règle, choisissez **Modifier les classifications** dans la liste déroulante **Classification** (**Objets** > **Règles de prévention des intrusions** > **Créer des règles** > **Modifier les classifications**).

Si **Afficher les classifications** s'affiche à la place, cela signifie que la configuration appartient à un domaine ascendant, ou que vous n'avez pas la permission de modifier la configuration.

Étape 2 Saisissez un **nom de classification** et une **description de classification**, comme décrit dans Détails des Événements liés aux intrusions, à la page 10.

Étape 3 Choisissez une priorité pour la classification dans la liste déroulante **Priority** (Priorité).

Étape 4 Cliquez sur **Add** (ajouter).

Étape 5 Cliquez sur **Done (Terminé)**.

Sujets connexes

Création de règles personnalisées, à la page 15

Définition d'une priorité d'événement

Procédure

Étape 1 Lors de la création ou de la modification d'une règle, choisissez la `priorité` dans la liste déroulante **Detection Options** (options de détection).

Étape 2 Cliquez sur **Add Option** (ajouter une option).

Étape 3 Choisissez une valeur dans la liste déroulante **Priorité**.

Étape 4 Cliquez sur **Save** (enregistrer).

Sujets connexes

Création de règles personnalisées, à la page 15

Définition d'une référence d'événement

Procédure

Étape 1 Lors de la création ou de la modification d'une règle, choisissez la `référence` dans la liste déroulante **Detection Options** (options de détection).

Étape 2 Cliquez sur **Add Option** (ajouter une option).

Étape 3 Saisissez une valeur dans le champ de **référence**, comme décrit dans Détails des Événements liés aux intrusions, à la page 10.

Étape 4 Cliquez sur **Save** (enregistrer).

Sujets connexes

Création de règles personnalisées, à la page 15

Création de règles personnalisées

Vous pouvez créer une règle de prévention des intrusions personnalisée comme suit :

- création de vos propres règles de texte standard
- enregistrement des règles de texte standard existantes en tant que nouvelles règles
- enregistrement des règles d'objet partagé fournies par le système comme nouvelles
- dans un déploiement multidomaine, enregistrement des règles ascendantes en tant que nouvelles règles dans un domaine descendant
- importation d'un fichier de règles local

Le système enregistre la règle personnalisée dans la catégorie de règle locale, quelle que soit la méthode que vous avez utilisée pour la créer.

Lorsque vous créez une règle de prévention des intrusions personnalisée, le système lui attribue un numéro de règle unique, qui a le format `GID:SID:Rev`. Les éléments composant ce numéro sont les suivants :

GID

ID de générateur Pour toutes les règles de texte standard, cette valeur est 1 (domaine global ou GID existant) ou 1000 à 2000 (domaines descendants). Pour toutes les règles d'objet partagé que vous enregistrez en tant que nouvelles, cette valeur est de 1.

SID

ID de Snort. Indique s'il s'agit d'une règle locale d'une règle système. Lorsque vous créez une règle, le système attribue le prochain SID disponible à une règle locale.

Les numéros SID des règles locales commencent à 1000000 et le SID de chaque nouvelle règle locale est incrémenté de un.

Rév.

Le numéro de révision. Pour une nouvelle règle, le numéro de révision est de 1. Chaque fois que vous modifiez une règle personnalisée, le numéro de révision est incrémenté de 1.

Dans une règle de texte standard personnalisée, vous définissez les paramètres d'en-tête de règle ainsi que les mots-clés et les arguments de la règle. Vous pouvez utiliser les paramètres d'en-tête de règle pour axer la règle de manière à ce qu'elle ne corresponde qu'au trafic utilisant un protocole spécifique et circulant vers ou à partir d'adresses IP ou de ports spécifiques.

Dans une règle de texte standard ou une règle d'objet partagé personnalisée fournie par le système, vous êtes limité à modifier les informations d'en-tête de règle telles que les ports source et de destination et les adresses IP. Vous ne pouvez pas modifier les mots-clés ou les arguments de la règle.

La modification des informations d'en-tête d'une règle d'objet partagé et l'enregistrement de vos modifications créent une nouvelle instance de la règle avec un ID de générateur (GID) de 1 (domaine global) ou de 1000 à 2000 (domaines descendants) et le prochain SID disponible pour une règle personnalisée . Le système lie la nouvelle instance de la règle d'objet partagé au mot-clé réservé `soid`, qui mappe la règle que vous créez à la

règle créée par Talos Intelligence Group. Vous pouvez supprimer des instances d'une règle d'objet partagé que vous créez, mais vous ne pouvez pas supprimer les règles d'objet partagé créées par Talos.

Rédaction de nouvelles règles

Procédure

Étape 1 Choisissez **Objects (objets)** > **Intrusion Rules (règles d'intrusion)**.

Étape 2 Cliquez sur **Create Rule** (créer une règle).

Étape 3 Saisissez une valeur dans le champ **Message**.

Étape 4 Choisissez une valeur dans chacune des listes déroulantes suivantes :

- **Classification**
- **Action**
- **Protocol** (Protocole)
- **Direction**

Étape 5 Saisissez des valeurs dans les champs suivants :

- **Source IPs (IP source)**
- **Destination IPs (IP de destination)**
- **Source Port (Port source)**
- **Destination Port (Port de destination)**

Le système utilise la valeur `any` si vous ne spécifiez aucune valeur pour ces champs.

Remarque Le système crée une carte réseau distincte pour chaque domaine enfant. Dans un déploiement multidomaine, l'utilisation d'adresses IP littérales pour limiter cette configuration peut avoir des résultats inattendus.

Étape 6 Choose a value from the **Detection Options** drop-down list.

Étape 7 Cliquez sur **Add Option** (ajouter une option).

Étape 8 Saisissez des arguments pour le mot-clé que vous avez ajouté.

Étape 9 Si vous le souhaitez, répétez les étapes 6 à 8.

Étape 10 Si vous avez ajouté plusieurs mots-clés, vous pouvez :

- Réorganiser les mots-clés – Cliquez sur la flèche vers le haut ou vers le bas à côté du mot-clé que vous souhaitez déplacer.
- Supprimer un mot-clé – Cliquez sur le **X** à côté de ce mot-clé.

Étape 11 Cliquez sur **Save As New** (Enregistrer comme nouveau).

Prochaine étape

- Activer vos règles nouvelles ou modifiées dans la politique de prévention des intrusions appropriée; voir Affichage des règles d'intrusion dans une politique d'intrusion.
- Déployer les changements de configuration.

Modification des règles existantes

Vous pouvez modifier les règles de prévention des intrusions personnalisées. Dans un déploiement multidomaine, vous pouvez modifier les règles de prévention des intrusions personnalisées qui appartiennent uniquement au domaine actuel.

Vous pouvez enregistrer les règles fournies par le système et les règles appartenant à des domaines ancêtres en tant que nouvelles règles personnalisées dans la catégorie de règles local, que vous pouvez ensuite modifier.

Procédure

Étape 1 Accédez aux règles de prévention des intrusions en utilisant l'une des méthodes suivantes :

- Choisissez **Policies (politiques)** > **Access Control (contrôle d'accès)** > **Intrusion**.

  Cliquez sur **Version Snort 2** à côté de la politique que vous souhaitez modifier et cliquez sur **Rules** (Règles).

- Choisissez **Objects (objets)** > **Intrusion Rules (règles d'intrusion)**.

Étape 2 Localisez la règle que vous souhaitez modifier. Vous avez les choix suivants :

- Parcourez les dossiers jusqu'à la règle.
- Recherchez la règle; voir Recherche de règles, à la page 19.
- Filtrer pour rechercher le groupe auquel la règle appartient; voir Règles de filtrage, à la page 24.

Étape 3 Cliquez sur **Edit** (✎) à côté de la règle ou, dans le cas de résultats de recherche, cliquez sur le message de règle.

Si **Afficher** (👁) apparaît plutôt, la configuration est héritée d'une politique ancêtre ou encore, vous n'êtes pas autorisé à modifier la configuration.

Étape 4 Modifiez la règle comme il convient pour le type de règle.

Remarque ne modifiez pas le protocole pour une règle d'objet partagé; dans le cas contraire, la règle serait inefficace.

Étape 5 Vous avez les choix suivants :

- Cliquez sur **Save** (Enregistrer) si vous modifiez une règle personnalisée et souhaitez remplacer la version actuelle de cette règle.
- Cliquez sur **Save As New** (Enregistrer comme nouvelle) si vous modifiez une règle fournie par le système ou une règle appartenant à un domaine ancêtre, ou si vous modifiez une règle personnalisée et que vous souhaitez enregistrer les modifications en tant que nouvelle règle.

Prochaine étape

- Si vous souhaitez utiliser la modification locale de la règle au lieu de la règle fournie par le système, désactivez la règle fournie par le système en utilisant les procédures figurant en États des règles d'intrusion et activez la règle locale.
- Déployer les changements de configuration.

Sujets connexes

Recherche de règles, à la page 19
Filtrage des règles dans la page de l'éditeur de règles de prévention des intrusions, à la page 21

Ajout de commentaires aux règles de prévention des intrusions

Vous pouvez ajouter des commentaires à n'importe quelle règle de prévention des intrusions. Ces commentaires peuvent être utiles pour fournir un contexte et des informations supplémentaires sur la règle et l'exploitation ou la violation de politique qu'elles identifient.

Dans un déploiement multidomaine, le système affiche les déploiements VPN créés dans le domaine actuel, que vous pouvez modifier. Il affiche également les règles créées dans les domaines ancêtres, que vous ne pouvez pas modifier. Pour afficher et modifier les règles créées dans un domaine inférieur, basculez vers ce domaine.

Procédure

Étape 1 Accédez aux règles de prévention des intrusions en utilisant l'une des méthodes suivantes :

- Choisissez **Policies (politiques)** > **Access Control (contrôle d'accès)** > **Intrusion**.

  Cliquez sur **Version Snort 2** à côté de la politique que vous souhaitez modifier et cliquez sur **Rules** (Règles).

- Choisissez **Objects (objets)** > **Intrusion Rules (règles d'intrusion)**.

Étape 2 Localisez la règle que vous souhaitez annoter. Vous avez les choix suivants :

- Parcourez les dossiers jusqu'à la règle.
- Recherchez la règle; voir Recherche de règles, à la page 19.
- Filtrer pour déterminer le groupe auquel la règle appartient; voir Règles de filtrage, à la page 24.

Étape 3 Cliquez sur **Edit** () à côté de la règle ou, dans le cas de résultats de recherche, cliquez sur le message de règle.

Si**Afficher** () apparaît à côté d'une règle, la règle appartient à une politique ancêtre ou vous n'êtes pas autorisé (e) à modifier la règle.

Étape 4 Cliquez sur **Commentaire sur la règle**.

Étape 5 Saisissez votre commentaire dans la zone de texte.

Étape 6 Cliquez sur **Add comment** (ajouter un commentaire).

Astuces Vous pouvez également ajouter et afficher des commentaires sur les règles dans l'affichage des paquets d'un incident d'intrusion.

Sujets connexes

Recherche de règles, à la page 19

Suppression de règles personnalisées

Vous pouvez supprimer des règles personnalisées si les règles ne sont pas actuellement activées dans une politique de prévention des intrusions. Vous ne pouvez pas supprimer les règles de texte standard ni les règles d'objet partagé fournies par le système. Dans un déploiement multidomaine, vous pouvez afficher et modifier les alertes du moniteur d'intégrité créées dans le domaine actuel uniquement.

Le système stocke les règles supprimées dans la catégorie supprimé, et vous pouvez utiliser une règle supprimée comme base pour une nouvelle règle. La page Rules (règles) d'une politique de prévention des intrusions n'affiche pas la catégorie supprimée, vous ne pouvez donc pas activer les règles personnalisées supprimées.

Astuces Les règles personnalisées comprennent les règles d'objets partagés que vous enregistrez avec les informations d'en-tête modifiées. Le système les enregistre également dans la catégorie de règle locale et les répertorie avec un GID de 1 (domaine global ou GID existant) ou de 1000 à 2000 (domaines descendants). Vous pouvez supprimer votre version modifiée d'une règle d'objet partagé, mais vous ne pouvez pas supprimer la règle d'objet partagé d'origine.

Procédure

Étape 1 Accédez aux règles de prévention des intrusions en utilisant l'une des méthodes suivantes :

- Choisissez **Policies (politiques)** > **Access Control (contrôle d'accès)** > **Intrusion**.

  Cliquez sur **Version Snort 2** à côté de la politique que vous souhaitez modifier et cliquez sur **Rules** (Règles).

- Choisissez **Objects (objets)** > **Intrusion Rules (règles d'intrusion)**.

Étape 2 Vous avez deux choix :

- Supprimer toutes les règles locales : cliquez sur **Delete Local Rules** (Supprimer les règles locales), puis sur **OK**.
- Supprimer une seule règle : choisissez `Local Rules (règles locales)` dans la liste déroulante **Group Rules By** (Grouper les règles par), cliquez sur **Supprimer** ( ) à côté de la règle que vous souhaitez supprimer, puis cliquez sur **OK** pour confirmer la suppression.

Sujets connexes

États des règles d'intrusion

Recherche de règles

Le système fournit des milliers de règles textuelles standard, et Talos Intelligence Group continue d'ajouter des règles à mesure que de nouvelles vulnérabilités et exploits sont découverts. Vous pouvez facilement rechercher des règles spécifiques pour pouvoir les activer, les désactiver ou les modifier.

Procédure

Étape 1 Accédez aux règles de prévention des intrusions en utilisant l'une des méthodes suivantes :

- Choisissez **Policies (politiques)** > **Access Control (contrôle d'accès)** > **Intrusion**.

  Cliquez sur **Version Snort 2** à côté de la politique que vous souhaitez modifier et cliquez sur **Rules** (Règles).

- Choisissez **Objects (objets)** > **Intrusion Rules (règles d'intrusion)**.

Étape 2 Cliquez sur **Search** (rechercher) dans la barre d'outils.

Étape 3 Ajoutez des critères de recherche

Étape 4 Cliquez sur **Search** (recherche).

Critères de recherche des règles de prévention des intrusions

Le tableau suivant décrit les options de recherche disponibles :

Tableau 7 : Critères de recherche de règle

Option	Description
ID de signature	Pour rechercher une seule règle basée sur ID de Snort (SID), saisissez un numéro SID. Pour rechercher plusieurs règles, saisissez une liste de numéros SID séparés par des virgules. Ce champ a une limite de 80 caractères.
ID de générateur	Pour rechercher des règles de texte standard, appuyez sur **1**. Pour rechercher des règles d'objet partagé, appuyez sur **3**.
Message	Pour rechercher une règle assortie d'un message précis, saisissez un seul mot du message relatif à la règle dans le champ **Message**. Par exemple, pour rechercher des exploits DNS, vous devez entrer `DNS`, ou pour rechercher des exploits de débordement de tampon, saisissez `overflow`.
Protocole	Pour rechercher des règles qui évaluent le trafic d'un protocole spécifique, sélectionnez le protocole. Si vous ne sélectionnez pas de protocole, les résultats de la recherche contiennent des règles pour tous les protocoles.
Source Port (port source)	Pour rechercher des règles qui inspectent les paquets provenant d'un port spécifié, saisissez un numéro de port source ou une variable liée au port.
Destination Port (port de destination)	Pour rechercher des règles qui inspectent les paquets destinés à un port spécifique, saisissez un numéro de port de destination ou une variable liée au port.
IP de la source	Pour rechercher des règles qui inspectent les paquets provenant d'une adresse IP spécifiée, saisissez une adresse IP source ou une variable liée à l'adresse IP.
IP de la destination	Pour rechercher des règles qui inspectent les paquets destinés à une adresse IP donnée, saisissez une adresse IP de destination ou une variable liée à l'adresse IP.

Option	Description
Mot-clé	Pour rechercher des mots-clés spécifiques, vous pouvez utiliser les options de recherche par mot-clé. Vous sélectionnez un mot-clé et saisissez une valeur de mot-clé à rechercher. Vous pouvez également faire précéder la valeur du mot-clé d'un point d'exclamation (`!`) pour correspondre à toute valeur autre que la valeur spécifiée.
Type	Pour rechercher des règles dans une catégorie spécifique, sélectionnez la catégorie dans la liste **Catégorie**.
Classification	Pour rechercher des règles qui ont une classification précise, sélectionnez le nom de la **classification** dans la liste Classification.
État de la règle	Pour rechercher des règles au sein d'une politique et d'un état de règle spécifiques, sélectionnez la politique dans la première liste **Rule State** (état de règle) et choisissez un état dans la deuxième liste pour rechercher les règles définies sur **Generate Events**, (Générer des événements) **Drop and Generate Events** (Abandonner et générer des événements) ou **Disabled** (Désactivé).

Filtrage des règles dans la page de l'éditeur de règles de prévention des intrusions

Vous pouvez filtrer les règles sur la page de l'éditeur de règles de prévention des intrusions pour afficher un sous-ensemble de règles. Cela peut être utile, par exemple, lorsque vous souhaitez modifier une règle ou son état, mais que vous avez de la difficulté à la trouver parmi les milliers de règles disponibles.

Lorsque vous saisissez un filtre, la page affiche tout dossier qui comprend au moins une règle correspondante ou un message lorsqu'aucune règle ne correspond.

Lignes directrices du filtrage

Votre filtre peut inclure des mots-clés spéciaux et leurs arguments, des chaînes de caractères et des chaînes de caractères littéraux entre guillemets, des espaces séparant plusieurs conditions de filtre. Un filtre ne peut pas inclure d'expressions régulières, de caractères génériques ni d'opérateur spécial tel qu'un caractère de négation (!), un symbole supérieur à (>), inférieur à (|), etc.

Tous les mots-clés, arguments de mots-clés et chaînes de caractères sont insensibles à la casse. À l'exception des mots-clés `gid` et `sid`, tous les arguments et toutes les chaînes sont traités comme des chaînes partielles. Les arguments pour `gid` et `sid` renvoient uniquement des correspondances exactes.

Vous pouvez développer un dossier sur la page d'origine non filtrée et le dossier reste développé lorsque le filtre suivant renvoie des correspondances dans ce dossier. Cela peut être utile lorsque la règle que vous souhaitez trouver se trouve dans un dossier qui contient un grand nombre de règles.

Vous ne pouvez pas limiter un filtre à un filtre ultérieur. Tout filtre que vous saisissez effectue une recherche dans l'ensemble de la base de données des règles et renvoie toutes les règles correspondantes. Lorsque vous saisissez un filtre alors que la page affiche toujours le résultat d'un filtre précédent, la page s'efface et renvoie le résultat du nouveau filtre à la place.

Vous pouvez utiliser les mêmes fonctionnalités avec des règles dans une liste filtrée ou non filtrée. Par exemple, vous pouvez modifier les règles d'une liste filtrée ou non filtrée sur la page de l'éditeur de règles de prévention des intrusions. Vous pouvez également utiliser l'une des options du menu contextuel de la page.

Astuces Le filtrage peut prendre beaucoup plus de temps lorsque le total combiné des règles de tous les sous-groupes est important, car les règles apparaissent dans plusieurs catégories, même lorsque le nombre total de règles uniques est beaucoup plus petit.

Filtrage par mots clés

Chaque filtre de règle peut inclure un ou plusieurs mots-clés au format :

```
keyword:argument
```

où mot-clé est l'un des mots-clés dans le tableau suivant et paramètre est une chaîne alphanumérique unique, insensible à la casse, à rechercher dans le champ spécifique ou les champs pertinents pour le mot-clé.

Les arguments pour tous les mots-clés, à l'exception de `gid` et `sid`, sont traités comme des chaînes partielles. Par exemple, l'argument `123` renvoie `"12345"`, `"41235"`, `"45123",` et ainsi de suite. Les arguments de `gid` et `sid` ne renvoient que des correspondances exactes; par exemple, `sid:3080` renvoie uniquement le SID 3080.

Astuces Vous pouvez rechercher un SID partiel en le filtrage avec une ou plusieurs chaînes de caractères.

Le tableau suivant décrit les mots-clés et les arguments de filtrage que vous pouvez utiliser pour filtrer les règles.

Tableau 8 : Mots-clés de filtres de règles

Mot-clé	Description	Exemple
`arachnids`	Renvoie une ou plusieurs règles en fonction de tout ou d'une partie de l'ID d'arachnides dans une référence de règle.	`arachnids:181`
`bugtraq`	Renvoie une ou plusieurs règles en fonction de tout ou d'une partie du Bugtraq ID dans une référence de règle.	`bugtraq:2120`
`cve`	Renvoie une ou plusieurs règles en fonction de tout ou d'une partie du numéro CVE dans une référence de règle.	`cve:2003-0109`
`gid`	L'argument `1` renvoie les règles de texte standard. L'argument `3` renvoie des règles d'objet partagé.	`gid:3`
`mcafee`	Renvoie une ou plusieurs règles en fonction de tout ou d'une partie de l'ID McAfee dans une référence de règle.	`mcafee:10566`
`msg`	Renvoie une ou plusieurs règles basées sur tout ou une partie du champ Message de la règle, également appelé message d'événement.	`msg:chat`
`nessus`	Renvoie une ou plusieurs règles basées sur tout ou une partie de l'ID Nessus dans une référence de règle.	`nessus:10737`

Mot-clé	Description	Exemple
`ref`	Renvoie une ou plusieurs règles basées sur tout ou une partie d'une chaîne alphanumérique unique dans une référence de règle ou dans le champ Message de la règle.	`ref:MS03-039`
`sid`	Renvoie la règle avec le ID de Snort exact.	`sid:235`
`url`	Renvoie une ou plusieurs règles basées sur tout ou une partie de l'URL dans une référence de règle.	`url:faqs.org`

Sujets connexes

Définition d'une référence d'événement, à la page 14
Détails des Événements liés aux intrusions, à la page 10

Filtrage des chaînes de caractères

Chaque filtre de règle peut inclure une ou plusieurs chaînes de caractères alphanumériques. Les chaînes de caractères recherchent le champ de **message** de règle, ID de Snort (SID) et l'ID de générateur (GID). Par exemple, la chaîne `123` renvoie les chaînes `"lotus123"`, `"123Mania"`et ainsi de suite dans le message de règle, et renvoie également SID 6123, SID 12375, etc.

Toutes les chaînes de caractères sont insensibles à la casse et sont traitées comme des chaînes partielles. Par exemple, les chaînes `ADMIN`, `admin` ou `Admin` renvoient `"admin"`, `"CFADMIN"`, `"Administrator"`, etc.

Vous pouvez mettre des chaînes de caractères entre guillemets pour renvoyer les correspondances exactes. Par exemple, la chaîne littérale `"overflow attempt"` entre guillemets ne renvoie que cette chaîne exacte, tandis qu'un filtre composé des deux chaînes `overflow` et `attempt` sans guillemets renvoie `"overflow attempt"`, `"overflow multipacket attempt"`, `"overflow with evasion attempt"`, et ainsi de suite.

Sujets connexes

Détails des Événements liés aux intrusions, à la page 10

Filtrage des combinaisons de mots-clés et de chaînes de caractères

Vous pouvez affiner les résultats du filtre en saisissant n'importe quelle combinaison de mots-clés, de chaînes de caractères ou des deux, séparés par des espaces. Le résultat inclut toute règle correspondant à toutes les conditions de filtre.

Vous pouvez saisir plusieurs conditions de filtre dans n'importe quel ordre. Par exemple, chacun des filtres suivants renvoie les mêmes règles :

- `url:at login attempt cve:200`
- `login attempt cve:200 url:at`
- `login cve:200 attempt url:at`

Règles de filtrage

Dans la page Règles de prévention des intrusions, vous pouvez filtrer les règles en sous-ensembles afin de pouvoir trouver plus facilement des règles spécifiques. Vous pouvez ensuite utiliser n'importe quelle fonctionnalité de la page, y compris en choisissant l'une des fonctionnalités disponibles dans le menu contextuel.

Le filtrage des règles peut être particulièrement utile pour localiser une règle spécifique à modifier.

Procédure

Étape 1 Accédez aux règles de prévention des intrusions en utilisant l'une des méthodes suivantes :

- Choisissez **Policies (politiques)** > **Access Control (contrôle d'accès)** > **Intrusion**.

  Cliquez sur **Version Snort 2** à côté de la politique que vous souhaitez modifier et cliquez sur **Rules** (Règles).

- Choisissez **Objects (objets)** > **Intrusion Rules (règles d'intrusion)**.

Étape 2 Avant le filtrage, vous avez les choix suivants:

- Développez tout groupe de règles que vous souhaitez développer. Certains groupes de règles ont également des sous-groupes que vous pouvez développer.

  Le développement d'un groupe sur la page d'origine non filtrée peut être utile lorsque vous vous attendez à ce qu'une règle se trouve dans ce groupe. Le groupe reste développé lorsque le filtre suivant génère une correspondance dans ce dossier et lorsque vous revenez à la page d'origine non filtrée en cliquant sur le filtre **Effacer** ( ).

- Choisissez une méthode de regroupement différente dans la liste déroulante **Group Rules By** (regrouper les règles par).

Étape 3 Saisissez les contraintes de filtre dans la zone de texte à côté de **Filtre** ( ) dans la liste **Group Rules By** (regrouper les règles par).

Étape 4 Appuyez sur Entrée.

Remarque Effacez la liste filtrée actuelle en cliquant sur le filtre **Effacer** ( ).

Mots clés et arguments dans les règles de prévention des intrusions

Le langage des règles vous permet de préciser le comportement d'une règle en combinant des mots-clés. Les mots clés et les valeurs associées (appelées *arguments*) dictent la façon dont le système évalue les paquets et les valeurs liées aux paquets qui sont testés par le moteur de règles. Le système Firepower prend actuellement en charge des mots-clés qui vous permettent d'effectuer des fonctions d'inspection, telles que la mise en correspondance de contenu, la mise en correspondance de modèles spécifiques au protocole et la mise en correspondance spécifique à un état. Vous pouvez définir jusqu'à 100 arguments par mot-clé et combiner

n'importe quel nombre de mots-clés compatibles pour créer des règles très spécifiques. Cela permet de réduire les risques de faux positifs et de faux négatifs et de cibler les renseignements sur les intrusions que vous recevez.

Notez que vous pouvez également utiliser Mises à niveau des profils adaptatifs dans les déploiements passifs pour adapter de manière dynamique le traitement actif des règles à des paquets spécifiques en fonction des métadonnées des règles et des informations sur l'hôte.

Les mots clés décrits dans cette section sont répertoriés sous les options de détection dans l'éditeur de règles.

Sujets connexes

À propos des profils adaptatifs

Les mots-clés content et protected_content

Utilisez le mot-clé `content` ou `protected_content` pour préciser le contenu que vous souhaitez détecter dans un paquet.

Vous devez presque toujours faire suivre un mot-clé `content` ou `protected_content` par des modificateurs qui indiquent où le contenu doit être recherché, si la recherche est sensible à la casse et d'autres options.

Notez que toutes les correspondances de contenu doivent être vraies pour que la règle déclenche un événement, c'est-à-dire que chaque correspondance de contenu entretient une relation ET avec les autres.

Notez également que, dans un déploiement en ligne, vous pouvez configurer des règles qui correspondent au contenu malveillant, puis le remplacer par votre propre chaîne de texte de longueur égale.

contenu

Lorsque vous utilisez le mot-clé `content`, le moteur de règles recherche cette chaîne dans la charge utile ou le flux du paquet. Par exemple, si vous saisissez `/bin/sh` comme valeur pour l'un des mots-clés `content`, le moteur de règles recherche dans la charge utile du paquet la chaîne `/bin/sh`.

Mettez en correspondance le contenu à l'aide d'une chaîne ASCII, d'un contenu hexadécimal (code d'octet binaire) ou d'une combinaison des deux. Entourez le contenu hexadécimal d'une barre verticale (|) dans la valeur du mot-clé. Par exemple, vous pouvez combiner du contenu hexadécimal et du contenu ASCII en utilisant quelque chose qui ressemble à `|90C8 C0FF FFFF|/bin/sh`.

Vous pouvez spécifier plusieurs correspondances de contenu dans une seule règle. Pour ce faire, utilisez des instances supplémentaires du mot-clé de `content`. Pour chaque correspondance de contenu, vous pouvez indiquer que des correspondances de contenu doivent être trouvées dans la charge utile ou le flux du paquet pour que la règle se déclenche.

Mise en garde Vous pouvez invalider votre politique de prévention des intrusions si vous créez une règle qui comprend un seul mot-clé de `content` et que l'option **Non** est sélectionnée pour ce mot-clé.

protected_content

Le mot-clé `protected_content` vous permet de coder la chaîne de contenu de votre recherche avant de configurer l'argument de règle. L'auteur de la règle d'origine utilise une fonction de hachage (SHA-512, SHA-256 ou MD5) pour encoder la chaîne avant de configurer le mot-clé.

Lorsque vous utilisez le mot-clé `protected_content` au lieu du mot-clé `content`, il n'y a aucun changement à la façon dont le moteur de règles recherche cette chaîne dans la charge utile ou le flux de paquet et la plupart des options de mots-clés fonctionnent comme prévu. Le tableau suivant résume les exceptions, pour lesquelles les options de mot-clé `protected_content` différent des options de mot-clé de `content`.

Tableau 9 : Exceptions d'options protected_content

Option	Description
Type de condensé	Nouvelle option pour le mot-clé de règle `protected_content`.
Insensible à la casse	Non pris en charge
Dans	Non pris en charge
Profondeur	Non pris en charge
Durée	Nouvelle option pour le mot-clé de règle `protected_content`.
Utiliser le sélecteur de motif rapide	Non pris en charge
sélecteur de motif rapide uniquement	Non pris en charge
Longueur et décalage du sélecteur de motif rapide	Non pris en charge

Cisco vous recommande d'inclure au moins un mot-clé de `content` dans les règles qui incluent un mot-clé `protected_content` pour s'assurer que le moteur de règles utilise l'analyseur de schéma rapide, ce qui accélère la vitesse de traitement et améliore les performances. Placez le mot-clé `content` avant le mot-clé `protected_content` dans la règle. Notez que le moteur de règles utilise la correspondance de modèle rapide lorsqu'une règle comprend au moins un mot-clé de `content`, que vous ayez ou non activé l'argument Use Fast Pattern Matcher (Utiliser un outil de recherche de motifs rapide) pour le mot-clé de `content`.

Mise en garde Vous pouvez invalider votre politique de prévention des intrusions si vous créez une règle qui comprend un seul mot-clé `protected_content` et que l'option **Non** est sélectionnée pour ce mot-clé.

Sujets connexes

Création de règles personnalisées, à la page 15
Arguments pour le contenu de base et le mot-clé protected_content, à la page 26
Le mot-clé replace, à la page 37

Arguments pour le contenu de base et le mot-clé protected_content

Vous pouvez restreindre l'emplacement et la sensibilité à la casse des recherches de contenu à l'aide de paramètres qui modifient le mot-clé « `content` » ou « `protected_content` ». Configurez les options qui modifient le mot-clé `content` ou `protected_content` pour spécifier le contenu que vous souhaitez rechercher.

Insensible à la casse

Remarque Cette option n'est **pas** prise en charge lors de la configuration du mot-clé `protected_content`.

Vous pouvez demander au moteur de règles d'ignorer la casse lors de la recherche de correspondances de contenu dans des chaînes ASCII. Pour que votre recherche ne soit pas sensible à la casse, cochez la **Insensible à la casse** lorsque vous spécifiez une recherche de contenu.

Type de condensé

Remarque Cette option **ne peut** être configurée qu'avec le mot-clé `protected_content`.

Utilisez le menu déroulant **Hash Type** pour identifier la fonction de hachage que vous avez utilisée pour encoder votre chaîne de recherche. Le système prend en charge le hachage SHA-512, SHA-256 et MD5 pour les chaînes de recherche `protected_content`. Si la longueur de votre contenu haché ne correspond pas au type de hachage sélectionné, le système n'enregistre **pas** la règle.

Le système sélectionne automatiquement la valeur par défaut définie par Cisco. Lorsque l' **option par défaut** est sélectionnée, aucune fonction de hachage n'est écrite dans la règle et le système utilise SHA-512 pour la fonction de hachage.

Données brutes

L'option **données brutes** indique au moteur de règles d'analyser la charge utile du paquet d'origine avant d'analyser les données de charge utile normalisées (décodées par une politique d'analyse de réseau) et n'utilise pas de valeur d'argument. Vous pouvez utiliser ce mot-clé lors de l'analyse du trafic Telnet pour vérifier les options de négociation Telnet dans la charge utile avant la normalisation.

Vous ne pouvez pas utiliser l'option de **données brutes** dans le même mot-clé `content` ou `protected_content` avec une option de contenu HTTP.

Astuces Vous pouvez configurer les options de **Profondeur du flux client** et de **Profondeur du flux serveur** du préprocesseur HTTP Inspect pour déterminer si les données brutes sont inspectées dans le trafic HTTP et quelle quantité de données brutes est inspectée.

Non

Sélectionnez l'option **Not** pour rechercher le contenu qui ne correspond pas au contenu spécifié. Si vous créez une règle qui comprend un mot-clé `content` ou `secure_content` avec l'option **Not** sélectionnée, vous devez également inclure dans la règle au moins un autre mot-clé `content` ou `protected_content` sans l'option **Not** sélectionnée.

Mise en garde Ne créez pas de règle qui comprend un seul mot-clé `content` ou `secure_content` si l'option **Not** (non) est sélectionnée pour ce mot-clé. Vous pourriez invalider votre politique de prévention des intrusions.

Par exemple, la règle SMTP 1:2541:9 comprend trois mots-clés `content`, dont l'option **Not** est sélectionnée. Une règle personnalisée basée sur cette règle ne serait pas valide si vous avez supprimé tous les mots-clés `content`, à l'exception de celui pour lequel l'option **Not** est sélectionnée. L'ajout d'une telle règle à votre politique de prévention des intrusions pourrait invalider la politique.

Astuces Vous ne pouvez pas sélectionner la case à cocher **Not** et la case à cocher **Use Fast Pattern Matcher** (Utiliser un outil de recherche de modèles rapide) avec le même mot-clé de `content`.

Emplacements de recherche du mot-clé protected_content et du contenu

Vous pouvez utiliser les options d'emplacement de recherche pour préciser où commencer la recherche du contenu précisé et jusqu'où la poursuivre.

combinaisons autorisées : arguments relatifs à l'emplacement de la recherche de contenu

Vous pouvez utiliser l'une ou l'autre de deux paires d'emplacements de `contenu` pour préciser où commencer la recherche du contenu et jusqu'où poursuivre la recherche, comme suit :

- Utilisez le **décalage** et la **profondeur** conjointement pour rechercher par rapport au début de la charge utile du paquet.
- Utilisez la **distance** et la **plage** conjointement pour rechercher par rapport à l'emplacement de recherche actuel.

Lorsque vous ne spécifiez qu'une seule option d'une paire, la valeur par défaut de l'autre option de la paire est utilisée.

Vous ne pouvez pas combiner les options de **décalage** et de **profondeur** avec les options de **distance** et de **plage**. Par exemple, vous ne pouvez pas associer un **décalage** et une **plage**. Vous pouvez utiliser n'importe quel nombre d'options d'emplacement dans une règle.

Lorsqu'aucun emplacement n'est précisé, les valeurs par défaut du **décalage** et de la **profondeur** sont utilisées; c'est-à-dire que la recherche de contenu commence au début de la charge utile du paquet et se poursuit jusqu'à la fin du paquet.

Vous pouvez également utiliser une variable `byte_extract` (extraction d'octets) existante pour spécifier la valeur d'une option d'emplacement.

Astuces Vous pouvez utiliser n'importe quel nombre d'options d'emplacement dans une règle.

Sujets connexes

Le mot-clé byte_extract, à la page 43

combinaisons autorisées : arguments relatifs à l'emplacement de la recherche du mot-clé protected_content

Utilisez l'option d'emplacement **Length** (Longueur) `protected_content` de pair avec l'option d'emplacement **Offset** (décalage) ou **Distance** pour préciser où commencer la recherche du contenu précisé et jusqu'où continuer la recherche, comme suit :

- Utilisez **Length** et **Offset** conjointement pour rechercher la chaîne protégée par rapport au début de la charge utile du paquet.

- Utilisez **Length** et **distance** ensemble pour rechercher la chaîne protégée par rapport à l'emplacement de la recherche actuelle.

Astuces Vous ne pouvez pas combiner les options **Offset** et **distance** dans une même configuration de mot-clé, mais vous pouvez utiliser n'importe quel nombre d'options d'emplacement dans une règle.

Lorsqu'aucun emplacement n'est spécifié, les valeurs par défaut sont utilisées; c'est-à-dire que la recherche de contenu commence au début de la charge utile du paquet et se poursuit jusqu'à la fin du paquet.

Vous pouvez également utiliser une variable `byte_extract` (extraction d'octets) existante pour spécifier la valeur d'une option d'emplacement.

Sujets connexes

Le mot-clé byte_extract, à la page 43

Arguments de l'emplacement de recherche content et protected_content

Profondeur

Remarque Cette option est **uniquement** prise en charge lors de la configuration du mot-clé `content`.

Spécifie la profondeur maximale de recherche de contenu, en octets, à partir du début de la valeur de décalage, ou si aucun décalage n'est configuré, à partir du début de la charge utile du paquet.

Par exemple, dans une règle avec une valeur de contenu `cgi-bin/phf`, une valeur de `décalage` de `3`et une valeur de `profondeur` de `22`, la règle commence à rechercher une correspondance avec la chaîne `cgi-bin/phf` à l'octet 3, et s'arrête après le traitement de 22 octets (octet 25) dans les paquets qui satisfont les paramètres spécifiés par l'en-tête de règle.

Vous devez spécifier une valeur supérieure ou égale à la longueur du contenu spécifié, jusqu'à un maximum de 65 535 octets. Vous ne pouvez pas indiquer la valeur 0.

La profondeur par défaut est pour la recherche jusqu'à la fin du paquet.

Distance

Demande au moteur de règles d'identifier les correspondances de contenu ultérieures qui se produisent un nombre spécifié d'octets après la précédente correspondance de contenu réussie.

Comme le compteur de distance commence à l'octet 0, spécifiez un de moins du nombre d'octets que vous souhaitez déplacer à partir de la dernière correspondance de contenu réussie. Par exemple, si vous spécifiez 4, la recherche commence au quatrième octet.

Vous pouvez spécifier une valeur de -65535 à 65535 octets. Si vous spécifiez une valeur de `Distance` négative, l'octet dans lequel vous commencez la recherche peut se trouver en dehors du début d'un paquet. Tous les calculs prendront en compte les octets à l'extérieur du paquet, même si la recherche commence en fait au premier octet du paquet. Par exemple, si l'emplacement actuel dans le paquet se trouve dans le cinquième octet et que l'option de règle de contenu suivante spécifie une valeur de `Distance` de -10 et une valeur de 20, dans l' `intervalle, la recherche commence au début de la charge utile et l'option Dans est` ajustée à 15.

La distance par défaut est de 0, ce qui signifie l'emplacement actuel dans le paquet après la dernière correspondance de contenu.

Durée

Remarque Cette option est **uniquement** prise en charge lors de la configuration du mot-clé `protected_content`.

L'option du mot-clé **Longueur** `protected_content` indique la longueur, en octets, de la chaîne de recherche sans lien.

Par exemple, si vous avez utilisé le contenu `Exemple1` pour générer un hachage sécurisé, utilisez `7` comme valeur de **longueur**. Vous **devez** saisir une valeur dans ce champ.

Décalage

Spécifie en octets, dans les données utiles du paquet, où commencer la recherche du contenu par rapport au début des données utiles du paquet. Vous pouvez spécifier une valeur de 65 535 à 65 535 octets.

Comme le compteur de décalage commence à l'octet 0, spécifiez un de moins du nombre d'octets que vous souhaitez déplacer à partir du début de la charge utile du paquet. Par exemple, si vous spécifiez 7, la recherche commence au neuvième octet.

Le décalage par défaut est de 0, ce qui signifie le début du paquet.

Dans

Remarque Cette option est **uniquement** prise en charge lors de la configuration du mot-clé `content`.

L'option **Within** (Dans) indique que, pour déclencher la règle, la prochaine correspondance de contenu doit se produire dans le nombre d'octets spécifié après la fin de la dernière correspondance de contenu réussie. Par exemple, si vous spécifiez une valeur **Within** de `8`, la prochaine correspondance de contenu doit se produire dans les huit octets suivants des données utiles du paquet, sinon elle ne répond pas aux critères qui déclenchent la règle.

Vous pouvez spécifier une valeur supérieure ou égale à la longueur du contenu spécifié, jusqu'à 65 535 octets.

La valeur par défaut pour **Within** est la recherche jusqu'à la fin du paquet.

Présentation : Contenu HTTP et arguments du mot-clé protected_content

Les options HTTP de mot-clés `content` ou `protected_content` vous permettent de spécifier où rechercher les correspondances de contenu dans un message HTTP décodé par le préprocesseur HTTP Inspect.

Deux options de champs d'état de recherche dans les réponses HTTP :

- **Code d'état HTTP**
- **Message d'état HTTP**

Notez que bien que le moteur de règles recherche dans les champs d'état bruts non normalisés, ces options sont répertoriées séparément pour simplifier l'explication ci-dessous des restrictions à prendre en compte lors de la combinaison d'autres champs HTTP bruts et normalisés.

Cinq options de recherche des champs normalisés dans les requêtes, les réponses HTTP ou les deux, selon les besoins :

- **URI HTTP**
- **Méthode HTTP**
- **En-tête HTTP**
- **Cookie HTTP**
- **Corps du client HTTP**

Trois options de recherche des champs bruts (non normalisés) sans état dans les requêtes, les réponses HTTP ou les deux, selon les besoins :

- **URI brut HTTP**
- **En-tête HTTP brut**
- **Cookie brut HTTP**

Utilisez les directives suivantes lors de la sélection des options `de contenu` HTTP :

- Les options de `contenu` HTTP s'appliquent uniquement au trafic TCP.
- Pour éviter un impact négatif sur les performances, sélectionnez uniquement les parties du message où le contenu spécifié peut s'afficher.

  Par exemple, lorsque le trafic est susceptible d'inclure des témoins volumineux comme ceux contenus dans des messages de panier d'achat, vous pouvez rechercher le contenu précisé dans l'en-tête HTTP, mais pas dans les témoins HTTP.

- Pour tirer parti de la normalisation du préprocesseur HTTP Inspect et améliorer les performances, toute règle liée à HTTP que vous créez doit au moins inclure un mot-clé `content` ou `protected_content` avec une **URI HTTP**,**une méthode HTTP**, **un en-tête HTTP** ou un **corps de client HTTP** sélectionné.
- Vous ne pouvez pas utiliser le mot-clé `replace` conjointement avec les options de mot-clé HTTP `content` ou `protected_content`.

Vous pouvez spécifier une seule option HTTP normalisée ou un seul champ d'état, ou utiliser des options HTTP normalisées et des champs d'état dans une combinaison quelconque pour cibler une zone de contenu à comparer. Cependant, notez les restrictions suivantes lors de l'utilisation des options de champ HTTP :

- Vous ne pouvez pas utiliser l'option de **données brutes** dans le même mot-clé `content` ou `protected_content` avec une option HTTP.
- Vous ne pouvez pas utiliser une option de champ (**URI brut HTTP**, **En-tête brut HTTP**, ou **Témoin brut HTTP**) ensemble avec le même mot-clé `content` ou `protected_content` avec son équivalent normalisé (**URI HTTP**, **En-tête HTTP**, ou **Témoin HTTP**, respectivement).
- Vous ne pouvez pas sélectionner **Use Fast Pattern Matcher** (Utiliser un outil de recherche de motifs rapide) avec une ou plusieurs des options de champ HTTP suivantes :

URI HTTP brut, **en-tête HTTP brut**, **témoin HTTP brut**, **témoin HTTP**, **méthode HTTP**, **message d'état HTTP** ou **code d'état HTTP**

Cependant, vous pouvez inclure les options ci-dessus dans un mot-clé `content` ou `protected_content` qui utilise également l'outil de recherche de schémas rapide pour rechercher l'un des champs normalisés suivants :

URI HTTP , **en-tête HTTP** ou **corps du client HTTP**

Par exemple, si vous sélectionnez **Témoin HTTP**, **En-tête HTTP** et **Use Fast Pattern Matcher** (Utiliser un outil de recherche de schéma rapide), le moteur de règles recherche le contenu à la fois dans le témoin HTTP et dans l'en-tête HTTP, mais l'outil de recherche de schéma rapide est appliqué uniquement à l'en-tête HTTP, pas au témoin HTTP.

- Lorsque vous combinez les options restreint et non restreint, la recherche de modèle rapide ne recherche que dans les champs non restreints que vous spécifiez pour tester s'il faut transmettre la règle à l'éditeur de règles de prévention des intrusions pour une évaluation complète, y compris l'évaluation des champs restreints.

Sujets connexes

Arguments de la recherche de schéma rapide pour mot-clé de contenu, à la page 35

Contenu HTTP et arguments du mot-clé protected_content

URI HTTP

Sélectionnez cette option pour rechercher des correspondances de contenu dans le champ URI de demande normalisée.

Notez que vous ne pouvez pas utiliser cette option avec l'option dd'URI HTTP (U) de mot-clé `pcre` pour rechercher le même contenu.

Remarque Un paquet de requête HTTP en pipeline contient plusieurs URI. Lorsque **HTTP URI**est sélectionné et que le moteur de règles détecte un paquet de requête HTTP en pipeline, le moteur de règles recherche dans tous les URI du paquet une correspondance de contenu.

URI brut HTTP

Sélectionnez cette option pour rechercher des correspondances de contenu dans le champ URI de demande normalisée.

Notez que vous ne pouvez pas utiliser cette option avec l'option dd'URI HTTP (U) de mot-clé `pcre` pour rechercher le même contenu.

Remarque Un paquet de requête HTTP en pipeline contient plusieurs URI. Lorsque **HTTP URI**est sélectionné et que le moteur de règles détecte un paquet de requête HTTP en pipeline, le moteur de règles recherche dans tous les URI du paquet une correspondance de contenu.

Méthode HTTP

Sélectionnez cette option pour rechercher des correspondances de contenu dans le champ de la méthode de demande, qui identifie l'action telle que GET et POST à entreprendre sur la ressource identifiée dans l'URI.

En-tête HTTP

Sélectionnez cette option pour rechercher les correspondances de contenu dans le champ d'en-tête normalisé, à l'exception des témoins, dans les requêtes HTTP. également dans les réponses lorsque l'option du préprocesseur HTTP Inspect **Inspect HTTP Responses** (Inspecter les réponses HTTP) est activée.

Notez que vous ne pouvez pas utiliser cette option avec l'option d'en-tête HTTP (H) du mot-clé `pcre` pour rechercher le même contenu.

En-tête HTTP brut

Sélectionnez cette option pour rechercher les correspondances de contenu dans le champ d'en-tête brut, à l'exception des témoins, dans les requêtes HTTP. également dans les réponses lorsque l'option du préprocesseur HTTP Inspect **Inspect HTTP Responses** (Inspecter les réponses HTTP) est activée.

Notez que vous ne pouvez pas utiliser cette option avec l'option d'en-tête HTTP brut (D) du mot-clé `pcre` pour rechercher le même contenu.

Cookie HTTP

Sélectionnez cette option pour rechercher des correspondances de contenu dans un témoin identifié dans un en-tête de demande client HTTP normalisé. Également dans les données set-cookie de la réponse lorsque l'option **Inspecter les réponses HTTP** du préprocesseur HTTP Inspect est activée. Notez que le système traite les témoins inclus dans le corps du message comme du contenu.

Vous devez activer l'option **Inspecter les témoins HTTP** du préprocesseur HTTP pour rechercher une correspondance uniquement dans le témoin ; sinon, le moteur de règles recherche dans l'en-tête entier, y compris le témoin.

Tenez compte des points suivants :

- Vous ne pouvez pas utiliser cette option en combinaison avec l'option mot-clé `pcre` de témoin HTTP (C) pour rechercher le même contenu.
- Les noms d'en-tête `Cookie` et `Set-Cookie :`, les espaces au début de la ligne d'en-tête et le `CRLF` qui termine la ligne d'en-tête sont inspectés dans le cadre de l'en-tête et non du témoin.

Cookie brut HTTP

Sélectionnez cette option pour rechercher des correspondances de contenu dans tout témoin identifié dans un en-tête de requête HTTP brute du client ; également dans les données set-cookie de la réponse lorsque l'option **Inspecter les réponses HTTP** du préprocesseur HTTP Inspect les est activée ; notez que le système traite les témoins inclus dans le corps du message comme le contenu du corps du message.

Vous devez activer l'option **Inspecter les témoins HTTP** du préprocesseur HTTP pour rechercher une correspondance uniquement dans le témoin ; sinon, le moteur de règles recherche dans l'en-tête entier, y compris le témoin.

Tenez compte des points suivants :

- Vous ne pouvez pas utiliser cette option avec l'option de témoin brut HTTP (K) de mot-clé `pcre` pour rechercher le même contenu.

- Les noms d'en-tête `Cookie` et `Set-Cookie :`, les espaces au début de la ligne d'en-tête et le `CRLF` qui termine la ligne d'en-tête sont inspectés dans le cadre de l'en-tête et non du témoin.

Corps du client HTTP

Sélectionnez cette option pour rechercher des correspondances de contenu dans le corps du message d'une requête client HTTP.

Notez que pour que cette option fonctionne, vous devez spécifier une valeur comprise entre 0 et 65 535 pour l'option de **profondeur d'extraction du corps du client HTTP du préprocesseur HTTP Inspect**.

Code d'état HTTP

Sélectionnez cette option pour rechercher les correspondances de contenu dans le code d'état à 3 chiffres dans une réponse HTTP.

Vous devez activer l'option **Inspecter les réponses HTTP** du préprocesseur HTTP Inspect pour que cette option renvoie une correspondance.

Message d'état HTTP

Sélectionnez cette option pour rechercher des correspondances de contenu dans la description textuelle qui accompagne le code d'état dans une réponse HTTP.

Vous devez activer l'option **Inspecter les réponses HTTP** du préprocesseur HTTP Inspect pour que cette option renvoie une correspondance.

Sujets connexes

Options du modificateur pcre, à la page 51
Options de normalisation HTTP au niveau du serveur

Vue d'ensemble : recherche de schéma rapide pour le mot-clé content

Remarque Ces options ne sont **pas** prises en charge lors de la configuration du mot-clé `protected_content`.

La l'analyseur rapide de schéma détermine rapidement quelles règles évaluer avant de transmettre un paquet au moteur de règles. Cette détermination initiale améliore les performances en réduisant considérablement le nombre de règles utilisées dans l'évaluation des paquets.

Par défaut, l'analyseur rapide de schémas recherche dans les paquets le contenu le plus long spécifié dans une règle. le but est d'éliminer le plus possible l'évaluation inutile d'une règle. Examinez l'exemple de fragment de règle suivant :

```
alert tcp any any -> any 80 (msg:"Exploit"; content:"GET";
http_method; nocase; content:"/exploit.cgi"; http_uri;
nocase;)
```

Pratiquement toutes les requêtes des clients HTTP contiennent le contenu `GET`, mais peu contiennent le contenu `/exploit.cgi`. L'utilisation de `GET` comme contenu de motif rapide amènerait le moteur de règles à évaluer cette règle dans la plupart des cas et aboutirait rarement à une correspondance. Cependant, la plupart des demandes `GET` des clients ne seraient pas évaluées à l'aide de `/exploit.cgi`, ce qui augmente les performances.

Le moteur de règles évalue le paquet par rapport à la règle uniquement lorsque l'analyseur rapide de schéma détecte le contenu spécifié. Par exemple, si un mot-clé `content` dans une règle spécifie le contenu `short`, un autre spécifie le contenu `long` et un troisième spécifie le contenu `le plus long`, l'outil de recherche de schéma rapide utilisera le contenu `le plus long` et la règle sera évaluée uniquement si le moteur de règles trouve `le plus long` dans la charge utile .

Arguments de la recherche de schéma rapide pour mot-clé de contenu

Utiliser le sélecteur de motif rapide

Cette option permet de spécifier un modèle de recherche plus court à utiliser par le moteur de recherche rapide Fast Pattern Matcher. Idéalement, le modèle que vous spécifiez est moins susceptible de se trouver dans le paquet que le modèle le plus long et, par conséquent, identifie plus spécifiquement l'exploitation ciblée.

Notez les restrictions suivantes lors de la sélection de l'option **Use Fast Pattern Matcher** et d'autres options dans le même mot-clé de `contenu` :

- Vous ne pouvez spécifier qu'une seule fois l'utilisation de **l'appariement rapide Fast Pattern Matcher** par règle.
- Vous ne pouvez pas utiliser **Distance**, **Within**, **Offset**, ou **Depth** lorsque vous sélectionnez **Use Fast Pattern Matcher** en combinaison avec **Not** (Non).
- Vous ne pouvez pas sélectionner Use Fast Pattern Matcher avec l'une des options de champ HTTP suivantes :

  URI HTTP brut, **en-tête HTTP brut**, **témoin HTTP brut**, **témoin HTTP**, **méthode HTTP**, **message d'état HTTP** ou **code d'état HTTP**

  Cependant, vous pouvez inclure les options ci-dessus dans un mot-clé `content` qui utilise également la correspondance de modèle rapide pour rechercher un des champs normalisés suivants :

  URI HTTP , **en-tête HTTP** ou **corps du client HTTP**

  Par exemple, si vous sélectionnez **Témoin HTTP**, **En-tête HTTP** et **Use Fast Pattern Matcher** (Utiliser un outil de recherche de schéma rapide), le moteur de règles recherche le contenu à la fois dans le témoin HTTP et dans l'en-tête HTTP, mais l'outil de recherche de schéma rapide est appliqué uniquement à l'en-tête HTTP, pas au témoin HTTP.

  Notez que vous ne pouvez pas utiliser une option de champ (**URI brut HTTP**, **En-tête brut HTTP**, ou **Témoin brut HTTP**) ensemble avec le même mot-clé `content` ou avec son équivalent normalisé (**URI HTTP**, **En-tête HTTP**, ou **Témoin HTTP**, respectivement).

  Lorsque vous combinez les options restreint et non restreint, le comparateur de modèle rapide recherche uniquement dans les champs non restreints que vous spécifiez pour tester s'il faut transmettre le paquet au moteur de règles pour une évaluation complète, y compris l'évaluation des champs restreints.

- Facultativement, lorsque vous sélectionnez **Use Fast Pattern matcher**, vous pouvez également sélectionner **Fast Pattern Matcher Only** ou **Fast Pattern Matcher Offset and Length** (Décalage et longueur de l'outil de recherche de modèles rapides), mais pas les deux.
- Vous ne pouvez pas utiliser l'appariement de modèle rapide lors de l'inspection de données Base64.

sélecteur de motif rapide uniquement

Cette option vous permet d'utiliser le mot-clé `content` uniquement comme option de recherche de modèle rapide et non comme option de règle. Vous pouvez utiliser cette option pour économiser les ressources

lorsqu'une évaluation par le moteur de règles du contenu précisé n'est pas nécessaire. Par exemple, envisageons un cas dans lequel une règle exige seulement que le contenu `12345` se trouve n'importe où dans les données utiles. Lorsque l'outil de recherche de modèle rapide détecte le modèle, le paquet peut être évalué par rapport à des mots-clés supplémentaires dans la règle. Le moteur de règles n'a pas besoin de réévaluer le paquet pour déterminer s'il comprend le modèle `12345`.

Vous n'utilisez pas cette option lorsque la règle contient d'autres conditions relatives au contenu spécifié. Par exemple, vous n'utiliserez pas cette option pour rechercher le contenu `1234` si une autre condition de règle cherche à déterminer si `abcd` se produit avant `1234`. Dans ce cas, le moteur de règles n'a pas pu déterminer l'emplacement relatif, car la spécification **de matcher de schéma rapide seulement** indique au moteur de règles de ne pas rechercher le contenu spécifié.

Tenez compte des conditions suivantes lorsque vous utilisez cette option :

- Le contenu précisé est indépendant de l'emplacement; c'est-à-dire qu'il peut se produire n'importe où dans la charge utile; par conséquent, vous ne pouvez pas utiliser les options de position (**Distance**, **Within**, **Offset**, **Depth** (Distance, entre, décalage, profondeur) ou **Fast Pattern Matcher Offset and Length**) (Décalage et longueur de l'outil de recherche de modèles rapides).
- Vous ne pouvez pas utiliser cette option avec **Not**.
- Vous ne pouvez pas utiliser cette option en combinaison avec **Décalage et longueur de l'outil de recherche de modèles rapides**.
- Le contenu spécifié sera traité comme insensible à la casse, car tous les schémas sont insérés dans le matcher de schémas rapide sans respecter la casse; cela est géré automatiquement, il n'est donc pas nécessaire de sélectionner **Insensible à la casse** lorsque vous sélectionnez cette option.
- Vous ne devez pas faire suivre immédiatement un mot-clé `content` qui utilise l'option **Fast Pattern Matcher Only** (Correspondance rapide de modèles uniquement) par les mots-clés suivants, qui définissent l'emplacement de la recherche par rapport à l'emplacement de la recherche actuelle :
  - `isdataat`
  - `pcre`
  - `content` lorsque **Distance** ou **Within** (à l'intérieur) est sélectionné
  - `content` lorsque l'**URI HTTP** est sélectionnée
  - `asn1`
  - `byte_jump`
  - `byte_test`
  - `byte_math`
  - `byte_extract`
  - `base64_decode`

Longueur et décalage du sélecteur de motif rapide

L'option **Fast Pattern Matcher Offset and Length** (Décalage et longueur de l'outil de recherche de modèles rapide) vous permet de spécifier une partie du contenu à rechercher. Cela peut réduire la consommation de mémoire dans les cas où le modèle est très long et seule une partie du modèle est suffisante pour identifier la

règle comme une correspondance probable. Lorsqu'une règle est sélectionnée par l'outil de recherche de modèles rapide, le modèle entier est évalué par rapport à la règle.

Vous déterminez la partie à utiliser par l'analyseur de modèle rapide en précisant en octets où commencer la recherche (décalage) et jusqu'où dans le contenu (longueur) à rechercher, en utilisant la syntaxe :

```
offset,length
```

Par exemple, pour le contenu :

```
1234567
```

si vous définissez le nombre d'octets de décalage et de longueur comme suit :

```
1,5
```

l'outil de recherche de modèles rapide ne recherche que le contenu `23456`.

Notez que vous ne pouvez pas utiliser cette option avec **Fast Pattern Matcher Only** (l'outil de recherche de modèles rapide uniquement).

Sujets connexes

Présentation : Contenu HTTP et arguments du mot-clé protected_content, à la page 30
Les mots-clés base64_decode et base64_data, à la page 119

Le mot-clé replace

Vous pouvez utiliser le mot-clé `replace` dans un déploiement en ligne pour remplacer le contenu spécifié ou pour remplacer le contenu dans le trafic SSL détecté par le Cisco SSL Appliance.

Pour utiliser le mot-clé `replace`, créez une règle de texte standard personnalisée qui utilise le mot-clé `content` pour rechercher une chaîne spécifique. Utilisez ensuite le mot-clé `replace` pour spécifier une chaîne pour remplacer le contenu. La valeur de remplacement et la valeur de contenu doivent être de même longueur.

Remarque Vous **ne pouvez pas** utiliser le mot-clé `replace` pour remplacer le contenu haché dans un mot-clé `protected_content`.

Vous pouvez également mettre la chaîne de remplacement entre guillemets pour assurer la compatibilité ascendante avec les versions précédentes du logiciel du système Firepower. Si vous n'incluez pas de guillemets, ils sont ajoutés à la règle automatiquement pour que la syntaxe de la règle soit correcte. Pour inclure un guillemet de début ou de fin dans le texte de remplacement, vous devez utiliser une barre oblique inverse pour le sortir, comme le montre l'exemple suivant :

```
"replacement text plus \"quotation\" marks""
```

Une règle peut contenir plusieurs mots-clés de `replace`, mais un seul par mot-clé `content`. Seule la première instance du contenu trouvé par la règle est remplacée.

Voici des exemples d'utilisation du mot-clé `replace` :

- Si le système détecte un paquet entrant qui contient un exploit, vous pouvez remplacer la chaîne malveillante par une autre sans danger. Parfois, cette technique réussit mieux que la simple suppression du paquet fautif. Dans certains scénarios d'attaque, l'agresseur renvoie simplement le paquet abandonné

jusqu'à ce qu'il contourne les défenses de votre réseau ou qu'il inonde votre réseau. En remplaçant les chaînes par une autre plutôt que d'abandonner le paquet, vous pouvez tromper l'agresseur en lui faisant croire que l'attaque a été lancée contre une cible qui n'était pas vulnérable.

- Si vous êtes confronté à des attaques de reconnaissance qui tentent de savoir si vous utilisez une version vulnérable, par exemple d'un serveur Web, vous pouvez détecter le paquet sortant et remplacer la bannière par votre propre texte.

Remarque Assurez-vous d'avoir défini l'état de la règle sur Générer des événements dans la politique de prévention des intrusions en ligne où vous souhaitez utiliser la règle de remplacement; définir la règle sur Drop (abandonner) et générer des événements entraînerait l'abandon du paquet, ce qui empêcherait le remplacement du contenu.

Dans le cadre du processus de remplacement de chaîne, le système met à jour automatiquement les sommes de contrôle des paquets afin que l'hôte de destination puisse recevoir le paquet sans erreur.

Notez que vous ne pouvez pas utiliser le mot-clé `replace` en combinaison avec les options de mot-clé `content` de message de requête HTTP.

Sujets connexes

Les mots-clés content et protected_content, à la page 25
Présentation : Contenu HTTP et arguments du mot-clé protected_content, à la page 30

Le mot-clé byte_jump

Le mot-clé `byte_jump` calcule le nombre d'octets définis dans un segment d'octets spécifié, puis saute ce nombre d'octets dans le paquet, soit en avant de la fin du segment d'octets spécifié, ou du début ou de la fin du paquet de charge, ou de un point par rapport à la dernière correspondance de contenu, selon les options que vous spécifiez. Il est utile dans les paquets où un segment spécifique d'octets décrit le nombre d'octets inclus dans les données variables au sein du paquet.

Le tableau suivant décrit les arguments requis par le mot-clé `byte_jump`.

Tableau 10 : Arguments byte_jump requis

Argument	Description
Octets	Nombre d'octets à extraire du paquet. S'il est utilisé sans DCE/RPC, les valeurs autorisées sont 0 à 10, avec les restrictions suivantes : • s'il est utilisé avec l'argument `de fin`, les octets peuvent avoir la valeur 0. Si Octets est 0, la valeur extraite est 0. • Si vous spécifiez un nombre d'octets autre que 1, 2 ou 4, vous devez spécifier un type de nombre (hexadécimal, octal ou décimal). Si elles sont utilisées avec DCE/RPC, les valeurs autorisées sont 1, 2 et 4.

Argument	Description
Décalage	Nombre d'octets dans la charge utile pour commencer le traitement Le compteur de `décalage` commence à l'octet 0, alors calculez la valeur de `décalage` en soustraire 1 du nombre d'octets que vous souhaitez faire sauter à partir du début de la charge utile du paquet ou de la dernière correspondance de contenu réussie. Vous pouvez définir -65 535 à 65 535 octets. Vous pouvez également utiliser une variable `byte_extract` existante ou un résultat `byte_math` pour spécifier la valeur de cet arguments.

Le tableau suivant décrit les options que vous pouvez utiliser pour définir comment le système interprète les valeurs que vous avez spécifiées pour les arguments requis.

Tableau 11 : Arguments byte_jump facultatifs supplémentaires

Argument	Description
Relatif	Rend le décalage relatif au dernier modèle trouvé dans la dernière correspondance de contenu réussie.
Harmoniser	Arrondit le nombre d'octets convertis à la limite supérieure de 32 bits.
Multiplicateur	Indique la valeur par laquelle le moteur de règles doit reproduire la valeur `byte_jump` obtenue à partir du paquet pour obtenir la valeur `byte_jump` finale. C'est-à-dire qu'au lieu de sauter le nombre d'octets définis dans un segment d'octets spécifié, le moteur de règles saute ce nombre d'octets multiplié par un entier que vous spécifiez avec l'argument Multiplicateur.
Décalage post-saut	Le nombre d'octets – 65 535 à 65 535 à sauter en avant ou en arrière après l'application d'autres arguments `byte_jump`. Une valeur positive fait faire un saut avant et une valeur négative en arrière. Laissez le champ vide ou saisissez `0` pour le désactiver. Notez que certains arguments `byte_jump` ne s'appliquent pas lorsque vous sélectionnez l'argument **DCE/RPC**.
Depuis le début	Indique que le moteur de règles doit ignorer le nombre spécifié d'octets dans la charge utile en commençant par le début de la charge utile du paquet, plutôt qu'à partir de la position actuelle dans le paquet.
De la fin	Le saut proviendra de l'octet qui suit le dernier octet de la mémoire tampon.
Masque binaire	Applique le masque de bits hexadécimal spécifié à l'aide de l'opérateur AND aux octets extraits de l'argument Bytes. Un masque de bits peut comporter de 1 à 4 octets. Le résultat sera déplacé vers la droite du nombre de bits égal au nombre de zéros à droite dans le masque.

Vous ne pouvez spécifier qu'un seul élément parmi **DCE/RPC**, **Endian** ou **Number Type**.

Si vous souhaitez définir comment le mot-clé `byte_jump` calcule les octets, vous pouvez choisir parmi les arguments décrits dans le tableau suivant. Si vous ne sélectionnez pas d'argument de classement des octets, le moteur de règles utilise l'ordre des octets au format big endian.

Tableau 12 : Arguments byte_jump concernant l'ordre des octets

Argument	Description
Gros-boutisme	Traite les données dans l'ordre des octets big endian, qui est l'ordre des octets par défaut du réseau.
Petit-boutisme	Traite les données dans l'ordre des octets little endian.
DCE/RPC	Spécifie un mot-clé `byte_jump` pour le trafic traité par le préprocesseur DCE/RPC. Le préprocesseur DCE/RPC détermine l'ordre des octets little ou big endian, et les arguments **type de numéro** et **Endian** ne s'appliquent pas. Lorsque vous activez cet arguments, vous pouvez également utiliser `byte_jump` conjointement avec d'autres mots-clés DCE/RPC spécifiques.

Définissez la façon dont le système affiche les données de chaîne dans un paquet à l'aide de l'un des arguments du tableau suivant.

Tableau 13 : Arguments du type de numéro

Argument	Description
Chaîne hexadécimale	Représente les données de chaîne converties au format hexadécimal.
Chaîne décimale	Représente les données de chaîne converties au format décimal.
Chaîne octale	Représente les données de chaîne converties au format octal.

Par exemple, si les valeurs que vous définissez pour `byte_jump` sont les suivantes :

- Octets = 4
- Décalage = 12
- Relatif activé
- Alignement activé

le moteur de règles calcule le nombre décrit dans les quatre octets qui apparaissent 13 octets après la dernière correspondance de contenu réussie et saute ce nombre d'octets dans le paquet. Par exemple, si les quatre octets calculés dans un paquet spécifique étaient `00 00 00 1F`, le moteur de règles le convertirait en 31. Comme `align` est spécifié (qui demande au moteur de se déplacer à la prochaine limite de 32 bits), le moteur de règles saute 32 octets dans le paquet.

Par ailleurs, si les valeurs que vous définissez pour `byte_jump` sont les suivantes :

- Octets = 4
- Décalage = 12

- À partir du début activé
- Multiplicateur = 2

le moteur de règles calcule le nombre décrit dans les quatre octets qui apparaissent 13 octets après le début du paquet. Ensuite, le moteur multiplie ce nombre par deux pour obtenir le nombre total d'octets à ignorer. Par exemple, si les quatre octets calculés dans un paquet spécifique étaient `00 00 00 1F`, le moteur de règles les convertirait en 31, puis les multiplie par deux pour obtenir 62. Comme l'appel du début est activé, le moteur de règles ignore les 63 premiers octets du paquet.

Sujets connexes

Le mot-clé byte_extract, à la page 43
Mots-clés DCE/RPC, à la page 77

Le mot-clé byte_test

Le mot-clé `byte_test` teste le segment d'octets spécifié en fonction de l'argument Value et de son opérateur.

Le tableau suivant décrit les arguments requis pour le mot-clé `byte_test`.

Tableau 14 : Arguments byte_test requis

Argument	Description
Octets	Le nombre d'octets à calculer à partir du paquet. Si elle est utilisée sans DCE/RPC, les valeurs autorisées sont comprises entre 1 et 10. Toutefois, si vous spécifiez un nombre d'octets autre que 1, 2 ou 4, vous devez spécifier un type de nombre (hexadécimal, octal ou décimal). Si elles sont utilisées avec DCE/RPC, les valeurs autorisées sont 1, 2 et 4.
Valeur	Valeur à tester, y compris son opérateur. Opérateurs pris en charge : &, ^, !>, !^, !=, !+ ou !^. Par exemple, si vous spécifiez `!1024`, `byte_test` convertit le nombre spécifié, et s'il n'était pas égal à 1024, il générerait un événement (si tous les autres paramètres clés correspondent). Notez que ! et != sont équivalentes. Vous pouvez également utiliser une variable `byte_extract` existante ou un résultat `byte_math` pour spécifier la valeur de cet arguments.
Décalage	Nombre d'octets dans la charge utile pour commencer le traitement Le compteur de `décalage` commence à l'octet 0, alors calculez la valeur de `décalage` en soustrayant 1 du nombre d'octets que vous souhaitez compter à partir du début de la charge utile du paquet ou de la dernière correspondance de contenu réussie. Vous pouvez utiliser une variable `byte_extract` ou un résultat `byte_math` pour spécifier la valeur de cet argument.

Vous pouvez définir plus en détail comment le système utilise les arguments `byte_test` avec les arguments décrits dans le tableau suivant.

Tableau 15 : Arguments byte_test facultatifs supplémentaires

Argument	Description
Masque binaire	Applique le masque de bits hexadécimal spécifié à l'aide de l'opérateur AND aux octets extraits de l'argument Bytes. Un masque de bits peut comporter de 1 à 4 octets. Le résultat sera déplacé vers la droite du nombre de bits égal au nombre de zéros à droite dans le masque.
Relatif	Établit le décalage par rapport à la dernière correspondance de modèle réussie.

Vous ne pouvez spécifier qu'un seul élément parmi **DCE/RPC**, **Endian** ou **Number Type**.

Pour définir comment le mot-clé `byte_test` calcule les octets qu'il teste, choisissez parmi les arguments du tableau suivant. Si vous ne sélectionnez pas d'argument de classement des octets, le moteur de règles utilise l'ordre des octets au format big endian.

Tableau 16 : Arguments byte_test pour l'ordre des octets

Argument	Description
Gros-boutisme	Traite les données dans l'ordre des octets big endian, qui est l'ordre des octets par défaut du réseau.
Petit-boutisme	Traite les données dans l'ordre des octets little endian.
DCE/RPC	Spécifie un mot-clé `byte_test` pour le trafic traité par le préprocesseur DCE/RPC. Le préprocesseur DCE/RPC détermine l'ordre des octets little ou big endian, et les arguments **type de numéro** et **Endian** ne s'appliquent pas. Lorsque vous activez cet arguments, vous pouvez également utiliser `byte_test` conjointement avec d'autres mots-clés DCE/RPC spécifiques.

Vous pouvez définir la façon dont le système affiche les données de chaîne dans un paquet en utilisant l'un des arguments du tableau suivant.

Tableau 17 : Arguments byte_test Type de numéro

Argument	Description
Chaîne hexadécimale	Représente les données de chaîne converties au format hexadécimal.
Chaîne décimale	Représente les données de chaîne converties au format décimal.
Chaîne octale	Représente les données de chaîne converties au format octal.

Par exemple, si la valeur de `byte_test` est spécifiée comme suit :

- Octets = 4
- Opérateur et valeur > 128
- Offset = 8

- Relatif activé

Le moteur de règles calcule le nombre décrit dans les quatre octets qui apparaissent à 9 octets de (par rapport à) la dernière correspondance de contenu réussie et, si le nombre calculé est supérieur à 128 octets, la règle est déclenchée.

Sujets connexes

Le mot-clé byte_extract, à la page 43
Mots-clés DCE/RPC, à la page 77

Le mot-clé byte_extract

Vous pouvez utiliser le mot-clé `byte_extract` pour lire un nombre spécifié d'octets d'un paquet dans une variable. Vous pouvez ensuite utiliser la variable ultérieurement dans la même règle comme valeur pour des arguments spécifiques dans certains autres mots-clés de détection.

Cela est utile, par exemple, pour extraire la taille des données de paquets où un segment spécifique d'octets décrit le nombre d'octets inclus dans les données du paquet. Par exemple, un segment spécifique d'octets pourrait indiquer que les données qui suivent comprennent quatre octets; vous pouvez extraire la taille de données de quatre octets pour les utiliser comme valeur de variable.

Vous pouvez utiliser `byte_extract` pour créer simultanément jusqu'à deux variables distinctes dans une règle. Vous pouvez redéfinir une variable `byte_extract` autant de fois que nécessaire; la saisie d'un nouveau mot-clé `byte_extract` avec le même nom de variable et une définition de variable différente remplace la définition précédente de cette variable.

Le tableau suivant décrit les arguments requis par le mot-clé `byte_extract`.

Tableau 18 : Arguments byte_extract nécessaires

Argument	Description
Octets à extraire	Nombre d'octets à extraire du paquet. Si vous spécifiez un nombre d'octets autre que 1, 2 ou 4, vous devez spécifier un type de nombre (hexadécimal, octal ou décimal).
Décalage	Le nombre d'octets dans la charge utile pour commencer l'extraction des données. Vous pouvez définir -65 535 à 65 535 octets. Le compteur de décalage commence à l'octet 0, alors calculez la valeur de décalage en soustraire 1 du nombre d'octets que vous souhaitez compter. Par exemple, spécifiez `7` pour compter vers l'avant 8 octets. Le moteur de règles compte vers l'avant à partir du début de la charge utile du paquet ou, si vous spécifiez également **Relative** (relatif), après la dernière correspondance de contenu réussie. Notez que vous pouvez spécifier des nombres négatifs uniquement lorsque vous spécifiez également **Relative**. Vous pouvez utiliser un résultat `byte_math` existant pour spécifier la valeur de cet arguments.
Nom de variable	Le nom de la variable à utiliser dans les arguments des autres mots-clés de détection. Vous pouvez spécifier une chaîne alphanumérique qui doit commencer par une lettre.

Pour définir plus précisément comment le système localise les données à extraire, vous pouvez utiliser les arguments décrits dans le tableau suivant.

Tableau 19 : Arguments byte_extract facultatifs supplémentaires

Argument	Description
Multiplicateur	Un multiplicateur pour la valeur extraite du paquet. Vous pouvez spécifier de 0 à 65 535. Si vous ne spécifiez pas de multiplicateur, la valeur par défaut est 1.
Harmoniser	Arrondit la valeur extraite à la valeur suivante de 2 ou 4 octets. Lorsque vous sélectionnez également **Multiplicateur**, le système applique le multiplicateur avant l'alignement.
Relatif	Rend le **décalage** relatif à la fin de la dernière correspondance de contenu réussie plutôt qu'au début de la charge utile.
Masque binaire	Applique le masque de bits hexadécimal spécifié à l'aide de l'opérateur AND aux octets extraits de l'argument Octets à extraire. Un masque de bits peut comporter de 1 à 4 octets. Le résultat sera déplacé vers la droite du nombre de bits égal au nombre de zéros à droite dans le masque.

Vous ne pouvez spécifier qu'un seul élément parmi **DCE/RPC**, **Endian** ou **Number Type**.

Pour définir comment le mot-clé `byte_extract` calcule les octets qu'il teste, vous pouvez choisir parmi les arguments du tableau suivant. Si vous ne sélectionnez pas d'argument de classement des octets, le moteur de règles utilise l'ordre des octets au format big endian.

Tableau 20 : Arguments byte_extract de l'ordre des octets

Argument	Description
Gros-boutisme	Traite les données dans l'ordre des octets big endian, qui est l'ordre des octets par défaut du réseau.
Petit-boutisme	Traite les données dans l'ordre des octets little endian.
DCE/RPC	Spécifie un mot-clé `byte_extract` pour le trafic traité par le préprocesseur DCE/RPC. Le préprocesseur DCE/RPC détermine l'ordre des octets little ou big endian, et les arguments **type de numéro** et **Endian** ne s'appliquent pas. Lorsque vous activez cet arguments, vous pouvez également utiliser `byte_extract` conjointement avec d'autres mots-clés DCE/RPC spécifiques.

Vous pouvez spécifier un type de numéro pour lire les données sous forme de chaîne ASCII. Pour définir la façon dont le système affiche les données de chaîne dans un paquet, vous pouvez sélectionner l'un des arguments dans le tableau suivant.

Tableau 21 : Arguments Type de nombre byte_extract

Argument	Description
Chaîne hexadécimale	Lit les données de chaîne extraites au format hexadécimal.

Argument	Description
Chaîne décimale	Lit les données de chaîne extraites au format décimal.
Chaîne octale	Lit les données de chaîne extraites au format octal.

Par exemple, si la valeur de `byte_extract` est spécifiée comme suit :

- Bytes to Extract = 4
- Nom de variable
- Offset = 8
- Relative = enabled

le moteur de règles lit le nombre décrit dans les quatre octets qui apparaissent à 9 octets de la dernière correspondance de contenu réussie dans une variable nommée `var`, que vous pouvez spécifier ultérieurement dans la règle comme valeur pour certains arguments de mots clés.

Le tableau suivant répertorie les arguments de mot-clé dans lesquels vous pouvez préciser une variable définie dans le mot-clé `byte_extract`.

Tableau 22 : Arguments de l'acceptation d'une variable byte_extract

Mot-clé	Argument
contenu	Profondeur, Décalage, Distance, Dans
byte_jump	Décalage
byte_test	Offset, Value
byte_math	RValue, Offset
isdataat	Décalage

Sujets connexes

Le préprocesseur DCE/RPC
Mots-clés DCE/RPC, à la page 77
Arguments pour le contenu de base et le mot-clé protected_content, à la page 26
Le mot-clé byte_jump, à la page 38
Le mot-clé byte_test, à la page 41
Caractéristiques des paquets, à la page 101

Le mot-clé byte_math

Le mot-clé `byte_math` effectue une opération mécanique sur une valeur extraite et une valeur spécifiée ou une variable existante, et stocke le résultat dans une nouvelle variable résultante. Vous pouvez ensuite utiliser la variable résultante comme arguments dans d'autres mots-clés.

Vous pouvez utiliser plusieurs mots-clés `byte_math` dans une règle pour effectuer plusieurs opérations byte_math.

Le tableau suivant décrit les arguments requis par le mot-clé `byte_math`.

Tableau 23 : Arguments byte_math requis

Argument	Description
Octets	Le nombre d'octets à calculer à partir du paquet. s'il est est utilisé sans DCE/RPC, les valeurs autorisées sont de 1 à 10 : • Les octets peuvent être compris entre 1 et 10 lorsque l'opérateur est +, -. *, ou /. • Les octets peuvent être compris entre 1 et 4 lorsque l'opérateur est <<or>>. • Si vous spécifiez un nombre d'octets autre que 1, 2 ou 4, vous devez spécifier un type de nombre (hexadécimal, octal ou décimal). Si elles sont utilisées avec DCE/RPC, les valeurs autorisées sont 1, 2 et 4.
Décalage	Nombre d'octets dans la charge utile pour commencer le traitement Le compteur de `décalage` commence à l'octet 0. Il faut donc calculer la valeur du `décalage` en soustrayant 1 du nombre d'octets que vous souhaitez avancer par rapport au début de la charge utile du paquet ou (si vous avez spécifié Relatif) par rapport à la dernière correspondance de contenu réussie. Vous pouvez définir -65 535 à 65 535 octets. Vous pouvez également spécifier la variable `byte_extract` ici.
Opérateur	+, -, *, /, <<, ou >>
RValue	La valeur après l'opérateur. Il peut s'agir d'un entier non signé ou d'une variable transmise par `byte_extract`.
Variable de résultat	Le nom de la variable dans laquelle le résultat du calcul `byte_math` sera stocké. Vous pouvez utiliser cette variable comme arguments dans d'autres mots-clés. Cette valeur est stockée sous la forme d'un entier non signé. Le nom de la variable : • Doit utiliser des caractères alphanumériques • Ne doit pas commencer par un chiffre • Peut inclure des caractères spéciaux pris en charge par la convention de nommage de fichier et de nom de variable Microsoft • Ne peut pas être entièrement constitué de caractères spéciaux

Le tableau suivant décrit les options que vous pouvez utiliser pour définir comment le système interprète les valeurs que vous avez spécifiées pour les arguments requis.

Tableau 24 : Arguments byte_math facultatifs supplémentaires

Argument	Description
Relatif	Fait en sorte que le décalage soit relatif au dernier schéma trouvé dans le dernier contenu réussi plutôt qu'au début de la charge utile.
Masque binaire	Applique le masque de bits hexadécimal spécifié à l'aide de l'opérateur AND aux octets extraits de l'argument Bytes. Un masque de bits peut comporter de 1 à 4 octets. Le résultat sera déplacé vers la droite du nombre de bits égal au nombre de zéros à droite dans le masque.

Vous ne pouvez spécifier qu'un seul élément parmi **DCE/RPC**, **Endian** ou **Number Type**.

Si vous souhaitez définir comment le mot-clé `byte_math` calcule les octets, vous pouvez choisir parmi les arguments décrits dans le tableau suivant. Si vous ne sélectionnez pas d'argument de classement des octets, le moteur de règles utilise l'ordre des octets au format big endian.

Tableau 25 : Arguments byte_math pour l'ordre des octets

Argument	Description
Gros-boutisme	Traite les données dans l'ordre des octets big endian, qui est l'ordre des octets par défaut du réseau.
Petit-boutisme	Traite les données dans l'ordre des octets little endian.
DCE/RPC	Spécifie un mot-clé `byte_math` pour le trafic traité par le préprocesseur DCE/RPC. Le préprocesseur DCE/RPC détermine l'ordre des octets little ou big endian, et les arguments **type de numéro** et **Endian** ne s'appliquent pas. Lorsque vous activez cet arguments, vous pouvez également utiliser `byte_math` avec d'autres mots-clés DCE/RPC spécifiques.

Définissez la façon dont le système affiche les données de chaîne dans un paquet à l'aide de l'un des arguments du tableau suivant.

Tableau 26 : Arguments du type de numéro

Argument	Description
Chaîne hexadécimale	Représente des données de chaîne au format hexadécimal.
Chaîne décimale	Représente les données de chaîne au format décimal.
Chaîne octale	Représente des données de chaîne au format octal.

Par exemple, si les valeurs que vous définissez pour `byte_math` sont les suivantes :

- Octets = 2
- Décalage = 0

- Opérateur = *
- RValue = hauteur
- Variable de résultat = zone

le moteur de règles extrait le nombre décrit dans les deux premiers octets du paquet et le multiplie par RValue (qui utilise la variable existante, `hauteur`) pour créer la nouvelle variable `zone`.

Tableau 27 : Arguments de l'acceptation d'une variable byte_math

Mot-clé	Argument
byte_jump	Décalage
byte_test	Offset, Value
byte_extract	Décalage
isdataat	Décalage

Présentation : le mot-clé pcre

Le mot-clé `pcre` vous permet d'utiliser des expressions régulières compatibles avec Perl (PCRE) pour inspecter les charges utiles des paquets pour rechercher du contenu spécifié. Vous pouvez utiliser PCRE pour éviter d'écrire plusieurs règles correspondant à de légères variations du même contenu.

Les expressions régulières sont utiles lors de la recherche de contenu qui pourrait être affiché de diverses manières. Le contenu peut avoir différents attributs que vous souhaitez prendre en compte dans votre tentative de le localiser dans la charge utile d'un paquet.

Notez que la syntaxe des expressions régulières utilisée dans les règles de prévention des intrusions est un sous-ensemble de la bibliothèque complète d'expressions régulières et diffère à certains égards de la syntaxe utilisée dans les commandes de la bibliothèque complète. Lorsque vous ajoutez un mot-clé `pcre` à l'aide de l'éditeur de règles de prévention des intrusions, saisissez la valeur complète au format suivant :

```
!/pcre/ ismxAEGRBUIPHDMCKSY
```

où :

- ! est une négation facultative (utilisez-la si vous souhaitez mettre en correspondance des modèles qui **ne correspondent pas** à l'expression régulière).
- /pcre/ est une expression régulière compatible avec Perl.
- `ismxAEGRBUIPHDMCKSY` est n'importe quelle combinaison d'options de modificateur.

Notez également que vous devez des caractères d'échappement répertoriés dans le tableau suivant pour que le moteur de règles les interprète correctement lorsque vous les utilisez dans une expression PCRE pour rechercher du contenu spécifique dans une charge utile de paquet.

Tableau 28 : Caractères PCRE échappés

Vous devez échapper...	avec une barre oblique inverse...	ou un code hexadécimal...
# (dièse)	\#	\x23
; (point-virgule)	\;	\x3B
| (barre verticale)	|	\x7C
: (deux-points)	\:	\x3A

Vous pouvez également utiliser `m?regex?`, où `?` est un délimiteur autre que /. Vous pouvez l'utiliser dans les cas où vous devez mettre en correspondance une barre oblique dans une expression régulière et que vous ne souhaitez pas y rajouter une barre oblique inverse. Par exemple, vous pourriez utiliser `m?`regex`?isxAEGRBUIPHDMCKSY`, où l'expression régulière est votre expression régulière compatible avec Perl et `ismxAEGRBUIPHDMCKSY` est une combinaison d'options de modificateur.

Astuces Vous pouvez éventuellement entourer votre expression régulière compatible avec Perl de guillemets, par exemple, pcre_expression ou `"` pcre_expression `"`. L'option d'utiliser des guillemets convient aux utilisateurs expérimentés familiarisés avec les versions précédentes lorsque les guillemets étaient obligatoires au lieu d'être optionnels. L'éditeur de règles de prévention des intrusions n'affiche pas de guillemets lorsque vous affichez une règle après l'avoir enregistrée.

Syntaxe pcre

Le mot-clé `pcre` accepte la syntaxe standard d'expression régulière compatible avec Perl (PCRE). Les sections suivantes décrivent cette syntaxe.

Astuces Bien que cette section décrit la syntaxe de base que vous pouvez utiliser pour PCRE, vous pouvez consulter une référence en ligne ou un livre dédié à Perl et PCRE pour des informations plus avancées.

Métacaractères

Les métacaractères sont des caractères littéraux qui ont une signification particulière dans les expressions régulières. Lorsque vous les utilisez dans une expression régulière, vous devez les faire précéder d'une « échappée » d'une barre oblique inverse.

Le tableau suivant décrit les métacaractères que vous pouvez utiliser avec PCRE et donne des exemples de chacun.

Tableau 29 : Métacaractères de PCRE

Métacaractère	Description	Exemple
.	Correspond à tous les caractères, à l'exception des retours à la ligne. Si `s` est utilisé comme option de modification, les caractères de retour à la ligne sont également inclus.	`abc.` correspond à `abcd`, `abc1`, `abc#`, etc.

Métacaractère	Description	Exemple
*	Ne correspond à aucune occurrence d'un caractère ou d'une expression.	abc* correspond à abc, abcc, abccc, abcccccc, etc.
?	Correspond à zéro ou une occurrence d'un caractère ou d'une expression.	abc? correspond à abc.
+	Correspond à une ou plusieurs occurrences d'un caractère ou d'une expression.	abc+ correspond à abc, abcc, abccc, abcccccc, etc.
()	Expressions de groupes.	(abc)+ correspond à abc, abcabcabc, abcabcabc et ainsi de suite.
{}	Spécifie une limite pour le nombre de correspondances pour un caractère ou une expression. Si vous souhaitez définir une limite inférieure et supérieure, séparez la limite inférieure et la limite supérieure par une virgule.	a{4,6}correspond à aaaa, aaaaa, ou aaaaaa. (ab) CLU correspond à aab.
[]	Vous permet de définir des classes de caractères et correspond à tout caractère ou combinaison de caractères décrit dans l'ensemble.	[abc123] correspond à a ou b ou c, et ainsi de suite.
^	Correspondance du contenu au début d'une chaîne. Également utilisé pour la négation, s'il est utilisé dans une classe de caractères.	^in correspond au « in » dans info, mais pas dans bac. [^a] trouve tout ce qui ne contient pas un.
$	Correspond au contenu à la fin d'une chaîne.	ce$ correspond au « ce » dans send, mais pas à cent.
|	Indique une expression OU.	(MAILTO|HELP) correspond à MAILTO ou HELP.
\	Vous permet d'utiliser des métacaractères comme caractères réels et est également utilisé pour spécifier une classe de caractères prédéfinie.	\. correspond à un point, * à un astérisque, \\ à une barre oblique inverse, etc. \d correspond aux caractères numériques, \w aux caractères alphanumériques, et ainsi de suite.

Classes de caractères

Les classes de caractères comprennent les caractères alphabétiques, les caractères numériques, les caractères alphanumériques et les espaces. Bien que vous puissiez créer vos propres classes de caractères entre parenthèses, vous pouvez utiliser les classes prédéfinies comme raccourcis pour différents types de types de caractères. Lorsqu'elle est utilisée sans qualificateur supplémentaire, une classe de caractères correspond à un seul chiffre ou caractère.

Le tableau suivant décrit et fournit des exemples de classes de caractères prédéfinies acceptées par PCRE.

Tableau 30 : Classes de caractères PCRE

Classes de caractères	Description	Définition de la classe de caractères
\d	Correspond à un caractère numérique (« chiffre »).	[0-9]

Classes de caractères	Description	Définition de la classe de caractères
\D	Correspond à tout ce qui n'est pas un caractère numérique.	[^0-9]
\w	Correspond à un caractère alphanumérique (« mot »).	[a-zA-Z0-9_]
\W	Correspond à tout ce qui n'est pas un caractère alphanumérique.	[a-zA-Z0-9_]
\s	Correspond aux espaces blancs, y compris les espaces, les retours, les tabulations, les retours à la ligne et les sauts de page.	[\r\t\n\f]
\S	Correspond à tout ce qui n'est pas un espace.	[^ \r\t\n\f]

Options du modificateur pcre

Vous pouvez utiliser les options de modification après avoir spécifié la syntaxe de l'expression régulière dans la valeur du mot-clé `pcre`. Ces modificateurs exécutent des fonctions de traitement propres à Perl, PCRE et à Snort. Les modificateurs apparaissent toujours à la fin de la valeur PCRE et se présentent dans le format suivant :

```
/pcre/ismxAEGRBUIPHDMCKSY
```

où `ismxAEGRBUPHMC` peut inclure n'importe quelle des options de modification figurant dans les tableaux suivants.

Astuces Vous pouvez éventuellement entourer l'expression régulière et toutes les options de modification de guillemets, par exemple, `"/pcre/ismxAEGRBUIPHDMCKSY"`. L'option d'utiliser des guillemets convient aux utilisateurs expérimentés familiarisés avec les versions précédentes lorsque les guillemets étaient obligatoires au lieu d'être optionnels. L'éditeur de règles de prévention des intrusions n'affiche pas de guillemets lorsque vous affichez une règle après l'avoir enregistrée.

Le tableau suivant décrit les options que vous pouvez utiliser pour effectuer les fonctions de traitement de Perl.

Tableau 31 : Options d'expression régulière Post liées à Perl

Option	Description
i	Rend l'expression régulière insensible à la casse.
s	Le point (.) décrit tous les caractères à l'exception du saut de ligne ou du caractère `\n`. Vous pouvez utiliser l'option `"s"` pour remplacer cela et faire en sorte que le point corresponde à tous les caractères, y compris le caractère de saut de ligne.
m	Par défaut, une chaîne est traitée comme une seule ligne de caractères, et `^` et `$` correspondent au début et à la fin d'une chaîne spécifique. Lorsque vous utilisez l'option `"m"`, `^` et `$` correspondent au contenu immédiatement avant ou après tout caractère de retour à la ligne dans la mémoire tampon, ainsi qu'au début ou à la fin de la mémoire tampon.

Option	Description
x	Ignore les espaces de données qui peuvent s'afficher dans le modèle, sauf lorsqu'ils sont protégés (précédés d'une barre oblique inverse) ou inclus dans une classe de caractères.

Le tableau suivant décrit les modificateurs PCRE que vous pouvez utiliser après l'expression régulière .

Tableau 32 : Options d'expression régulière Post Liées à PCRE

Option	Description
A	Le modèle doit correspondre au début de la chaîne (identique à l'utilisation de `^` dans une expression régulière).
E	Définit `$` pour qu'il corresponde uniquement à la fin de la chaîne d'objet. (Sans `E`, `$` correspond également immédiatement avant le caractère final s'il s'agit d'un retour à la ligne, mais pas avant d'autres caractères de nouvelle ligne.)
G	Par défaut, `*` `+` et `?` sont « greedy », ce qui signifie que si deux correspondances ou plus sont trouvées, ils choisiront la correspondance la plus longue. Utilisez le caractère G pour modifier ce réglage afin que ces caractères choisissent toujours la première correspondance, sauf si elle est suivie d'un point d'interrogation (`?`). Par exemple, `*?` `?` et `??` serait gourmand en ressources dans une construction utilisant le modificateur G et toute incidence de `*`, `+`ou `?` sans le point d'interrogation supplémentaire ne le serait pas.

Le tableau suivant décrit les modificateurs spécifiques à Snort que vous pouvez utiliser après l'expression régulière.

.

Tableau 33 : Modificateurs d'expression régulière Post spécifiques à Snort

Option	Description
R	Recherche le contenu correspondant par rapport à la fin de la dernière correspondance trouvée par le moteur de règles.
B	Recherche le contenu dans les données avant qu'il ne soit décodé par un préprocesseur (cette option revient à utiliser l'argument de `données brutes` avec le mot-clé `content` ou `protected_content`).
U	Recherche le contenu de l'URI d'un message de requête HTTP normalisé décodé par le préprocesseur HTTP Inspect. Notez que vous ne pouvez pas utiliser cette option avec l'option d'**URI HTTP**`content` ou `protected_content` du mot clé pour rechercher le même contenu. Notez qu'un paquet de requête HTTP en pipeline contient plusieurs URI. Une expression PCRE qui inclut l'option U permet au moteur de règles de rechercher une correspondance de contenu uniquement dans le premier URI d'un paquet de requête HTTP en pipeline. Pour rechercher tous les URI du paquet, utilisez le mot-clé `content` ou `protected_content` avec l'**URI HTTP** sélectionné, avec ou sans une expression PCRE qui utilise l'option U.

Option	Description
I	Recherche le contenu de l'URI d'un message de requête HTTP brut décodé par le préprocesseur HTTP Inspect. Notez que vous ne pouvez pas utiliser cette option avec l'option de mot-clé `content` ou `protected_content` de **HTTP Raw URI** pour rechercher le même contenu
P	Recherche le contenu dans le corps d'un message de requête HTTP normalisé décodé par le préprocesseur HTTP Inspect.
H	Recherche le contenu de l'en-tête, à l'exception des témoins, d'une requête HTTP ou d'un message de réponse décodé par le préprocesseur HTTP Inspect. Notez que vous ne pouvez pas utiliser cette option avec l'option de mot-clé `content` ou `protected_content` de l'**En-tête HTTP** pour rechercher le même contenu.
D	Recherche le contenu de l'en-tête, à l'exception des témoins, d'une requête HTTP brute ou d'un message de réponse décodé par le préprocesseur HTTP Inspect. Notez que vous ne pouvez pas utiliser cette option avec l'option de mot-clé `content` ou `protected_content` de l'**En-tête brut HTTP** pour rechercher le même contenu.
L	Recherche le contenu du champ de méthode d'un message de requête HTTP normalisé décodé par le préprocesseur HTTP Inspect; le champ de méthode identifie l'action telle que GET, PUT, CONNECT, etc., à entreprendre sur la ressource identifiée dans l'URI.
C	Lorsque l'option HTTP Inspect préprocesseur **Inspect HTTP cookies** est activée, recherche le contenu normalisé dans tout témoin dans un en-tête de requête HTTP, ainsi que dans tout set-cookie dans un en-tête de réponse HTTP lorsque l'option préprocesseur **Inspect HTTP Responses** est activée. Lorsque **Inspect HTTP cookies** n'est pas activé, recherche dans l'ensemble de l'en-tête, y compris les données de témoin ou set-cookie. Tenez compte des points suivants : • Les témoins inclus dans le corps du message sont traités comme du contenu. • Vous ne pouvez pas utiliser cette option avec l'option de mot-clé `content` ou `protected_content` de **Témoin HTTP** pour rechercher le même contenu. • Les noms d'en-tête `Cookie` et `Set-Cookie :`, les espaces au début de la ligne d'en-tête et le `CRLF` qui termine la ligne d'en-tête sont inspectés dans le cadre de l'en-tête et non du témoin.

Option	Description
K	Lorsque l'option HTTP Inspect préprocesseur **Inspect HTTP cookies** est activée, recherche le contenu brut de tout témoin dans un en-tête de requête HTTP, ainsi que de tout set-cookie dans un en-tête de réponse HTTP lorsque l'option de préprocesseur **Inspect HTTP Responses** est activée. Lorsque **Inspect HTTP cookies** n'est pas activé, recherche dans l'ensemble de l'en-tête, y compris les données de témoin ou set-cookie. Tenez compte des points suivants : • Les témoins inclus dans le corps du message sont traités comme du contenu. • Vous ne pouvez pas utiliser cette option avec l'option de mot-clé `content` ou `protected_content` de **Témoin brut HTTP** pour rechercher le même contenu. • Les noms d'en-tête `Cookie` et `Set-Cookie :`, les espaces au début de la ligne d'en-tête et le `CRLF` qui termine la ligne d'en-tête sont inspectés dans le cadre de l'en-tête et non du témoin.
S	Recherche le code d'état à trois chiffres dans une réponse HTTP.
O	Recherche la description textuelle qui accompagne le code d'état dans une réponse HTTP.

Remarque N'utilisez pas l'option U conjointement avec l'option R. Cela pourrait entraîner des problèmes de performances. De plus, n'utilisez pas l'option U en combinaison avec une autre option de contenu HTTP (I, P, H, D, M, C, K, S ou Y).

Sujets connexes

Présentation : Contenu HTTP et arguments du mot-clé protected_content, à la page 30

Exemples de valeurs de mot clé pcre

Les exemples suivants montrent des valeurs que vous pourriez saisir pour `pcre`, avec des descriptions de ce à quoi chaque exemple correspondrait.

- **`/feedback[(\d{0,1})]?\.cgi/U`**

Cet exemple recherche dans la charge utile d'un paquet `les commentaires`, suivi de zéro ou un caractère numérique, suivi de `.cgi` et situé uniquement dans les données URI.

Cet exemple correspondrait à :

- `feedback.cgi`
- `feedback1.cgi`
- `feedback2.cgi`
- `feedback3.cgi`

Cet exemple ne correspondrait **pas** à :

- `feedbacka.cgi`
- `feedback11.cgi`
- `feedback21.cgi`
- `feedbackzb.cgi`

- **`/^ez(\w{3,5})\.cgi/iU`**

Cet exemple recherche la charge utile d'un paquet `ez` au début d'une chaîne, suivi d'un mot de 3 à 5 lettres, suivi de `.cgi`. La recherche ne respecte pas la casse et ne recherche que les données URI.

Cet exemple correspondrait à :

- `EZBoard.cgi`
- `ezman.cgi`
- `ezadmin.cgi`
- `EZAdmin.cgi`

Cet exemple ne correspondrait **pas** à :

- `ezez.cgi`
- `fez.cgi`
- `abcezboard.cgi`
- `ezboardman.cgi`

- **`/mail(file|seek)\.cgi/U`**

Cet exemple permet de rechercher `mail`, suivi de `file` ou `seek`, dans des données URI.

Cet exemple correspondrait à :

- `mailfile.cgi`
- `mailseek.cgi`

Cet exemple ne correspondrait **pas** à :

- `MailFile.cgi`
- `mailfilefile.cgi`

- **`m?http\\x3a\x2f\x2f.*(\n|\t)+?U`**

Cet exemple recherche dans la charge utile des paquets le contenu de l'URI pour un caractère de tabulation ou de nouvelle ligne dans une requête HTTP, après n'importe quel nombre de caractères. Cet exemple utilise `m?`regex`?` pour éviter d'utiliser `http\:\/\/` dans l'expression. Notez que les deux-points sont précédés d'une barre oblique inverse.

Cet exemple correspondrait à :

- `http://www.example.com?scriptvar=x&othervar=\n\..\..`

- `http://www.example.com?scriptvar=\t`

Cet exemple ne correspondrait **pas** à :

- `ftp://ftp.example.com?scriptvar=&othervar=\n\..\..`
- `http://www.example.com?scriptvar=|/bin/sh -i|`
- **`m?http\\x3a\x2f\x2f.*=\|.*\|+?sU`**

Cet exemple recherche dans la charge utile du paquet une URL contenant un nombre quelconque de caractères, y compris des nouvelles lignes, suivies d'un signe égal, et des caractères de type barre verticale contenant un nombre quelconque de caractères ou d'espaces blancs. Cet exemple utilise `m?`regex`?` pour éviter d'utiliser `http\:\/\/` dans l'expression.

Cet exemple correspondrait à :

- `http://www.example.com?value=|/bin/sh/ -i|`
- `http://www.example.com?input=|cat /etc/passwd|`

Cet exemple ne correspondrait **pas** à :

- `ftp://ftp.example.com?value=|/bin/sh/ -i|`
- `http://www.example.com?value=x&input?|cat /etc/passwd|`
- **`/[0-9a-f]{2}\:[0-9a-f]{2}\:[0-9a-f]{2}\:[0-9a-f]{2}\:[0-9a-f]{2}\:[0-9a-f]{2}/i`**

Cet exemple permet de rechercher n'importe quelle adresse MAC dans la charge utile de paquets. Notez qu'il échappe aux caractères deux-points par des barres obliques inverses.

Le mot-clé metadata

Vous pouvez utiliser le mot-clé `metadata` (métadonnées) pour ajouter vos propres informations descriptives à une règle. Vous pouvez également utiliser le mot-clé `métadonnées` avec des arguments de `service` pour identifier les applications et les ports dans le trafic réseau. Vous pouvez utiliser les informations que vous ajoutez pour organiser ou identifier les règles de la manière qui vous convient. Vous pouvez rechercher dans les règles les informations que vous ajoutez et les arguments de `service`.

Le système valide les métadonnées en fonction du format de l'argument :

key value

où *key* (clé) et *value* (valeur) fournissent une description combinée séparée par une espace. Il s'agit du format utilisé par Talos Intelligence Group pour ajouter des métadonnées aux règles fournies par Cisco.

Vous pouvez également utiliser le format :

key = value

Par exemple, vous pouvez utiliser le format de *valeur de clé* pour identifier les règles par auteur et date, en utilisant une catégorie et une sous-catégorie comme suit :

```
author SnortGuru_20050406
```

Vous pouvez utiliser plusieurs mots-clés `metadata` (métadonnées) dans une règle. Vous pouvez également utiliser des virgules pour séparer plusieurs arguments *valeur de clé* dans un seul mot-clé `métadonnées`, comme le montre l'exemple suivant :

```
author SnortGuru_20050406, revised_by SnortUser1_20050707,
revised_by SnortUser2_20061003,
revised_by SnortUser1_20070123
```

Vous n'êtes pas limité à utiliser un format *key value* ou *key=value* Cependant, vous devez être conscient des limites résultant de la validation basée sur ces formats.

Caractères restreints à éviter

Notez les restrictions suivantes concernant les caractères :

- N'utilisez pas de point-virgule (;) ni de deux-points (:).
- Le système interprète une virgule comme séparateur pour plusieurs arguments valeur de *key value* ou *key=value*. Par exemple :

  key value, *key value*, *key value*
- Le système interprète le signe égal (=) ou l'espace comme un séparateur entre *la clé* et *la valeur*. Par exemple :

  key value

  key=value

Tous les autres caractères sont autorisés.

Métadonnées réservées à éviter

Évitez d'utiliser les mots suivants dans un mot-clé de `métadonnées`, comme arguments uniques ou comme *clé* dans un *paramètre clé-valeur* : ceux-ci sont réservés à l'usage de Talos :

```
application
engine
impact_flag
os
policy
rule-type
rule-flushing
soid
```

Remarque Communiquez avec le service d'assistance pour obtenir de l'aide sur l'ajout de métadonnées restreintes aux règles locales qui pourraient ne pas fonctionner comme prévu.

Niveau d'incidence 1

Vous pouvez utiliser l'argument de *valeur de clé* réservée suivant dans un mot-clé de `métadonnées` :

```
impact_flag red
```

Cet argument *clé-valeur* définit l'indicateur d'impact à rouge (niveau 1) pour une règle locale que vous importez ou une règle personnalisée que vous créez à l'aide de l'éditeur de règles de prévention des intrusions.

Notez que lorsque Talos inclut l'argument `impact_flag_red` dans une règle fournie par Cisco, Talos a déterminé qu'un paquet déclenchant la règle indique que l'hôte source ou de destination est potentiellement altéré par un virus, un cheval de Troie ou un autre logiciel malveillant.

Métadonnées de service

Le système détecte les applications en cours d'exécution sur les hôtes de votre réseau et insère des informations de protocole d'application dans votre trafic réseau. il le fait quelle que soit la configuration de votre politique de découverte. Vous pouvez utiliser des arguments de `service` de mots-clés de `métadonnées` dans une règle TCP ou UDP pour faire correspondre les protocoles d'application et les ports au sein de votre trafic réseau. Vous pouvez combiner un ou plusieurs arguments d'application de `service` en une règle avec un seul arguments de port.

Applications de service

Vous pouvez utiliser le mot-clé `métadonnées` avec `service` comme *clé* et une application comme *valeur* pour faire correspondre les paquets au protocole d'application identifié. Par exemple, l'argument *valeur de clé* suivant dans un mot-clé de `métadonnées` associe la règle au trafic HTTP :

```
service http
```

Vous pouvez identifier plusieurs applications séparées par des virgules. Par exemple :

```
service http, service smtp, service ftp
```

Mise en garde Le profilage adaptatif **doit** être activé (son état par défaut) comme décrit dans Configuration des profils adaptatifs pour que les règles de prévention des intrusions utilisent les métadonnées de service.

Le tableau suivant décrit les valeurs d'application les plus couramment utilisées avec le mot-clé `service`.

Remarque Contactez le service d'assistance si vous avez de la difficulté à identifier des applications qui ne figurent pas dans le tableau.

Tableau 34 : Valeurs de service

Valeur	Description
`cvs`	Système de versions simultanées
`dcerpc`	Système d'environnement informatique distribuée/appels de procédure à distance
`dns`	Domain Name System (DNS, système de nom de domaine)
`finger`	Protocole d'information sur les utilisateurs de doigts
`ftp`	Protocole de transfert de fichier (File Transfer Protocol)
`données-ftp`	Protocole de transfert de fichier (File Transfer Protocol)

Valeur	Description
`http`	Protocole de transfert hypertexte (HyperText Transfer Protocol)
`imap`	protocole IMAP (Internet Message Access Protocol)
`isakmp`	protocole ISAKMP (Internet Security Association and Key Management Protocol)
`mysql`	Mon langage de requête structuré
`netbios-dgm`	Service de datagramme NETBIOS
`netbios-ns`	Service de nom NETBIOS
`netbios-ssn`	Service de session NETBIOS
`nntp`	Protocole de transfert des informations du réseau
`oracle`	Services réseau Oracle
`shell`	Shell de système d'exploitation
`pop2`	Protocole du bureau de poste, version 2
`pop3`	Protocole du bureau de poste, version 3
`smtp`	Protocole de transfert de messagerie simple
`snmp`	Protocole SNMP (gestion de réseau simple)
`ssh`	Protocole réseau Secure Shell
`sunrpc`	Protocole d'appel de procédure à distance Sun
`telnet`	Protocole de réseau Telnet
`tftp`	protocole TFTP (Trivial File Transfer Protocol)
`x11`	Système X Window

Ports de service

Vous pouvez utiliser le mot-clé `métadonnées` avec `service` comme *clé* et un arguments de port spécifiés comme *valeur* pour définir comment la règle correspond aux ports en combinaison avec les applications.

Vous pouvez spécifier n'importe quelle valeur de port dans le tableau ci-dessous, une valeur par règle.

Tableau 35 : Valeurs de port de service

Valeur	Description
`else-ports` ou `unknown`	Le système applique la règle si l'une des conditions suivantes est remplie : • L'application du paquet est connue et correspond à l'application de la règle. • L'application de paquets est inconnue et les ports de paquets correspondent aux ports de la règle. Les valeurs « `else-ports` » et « `unknown` » produisent le comportement par défaut que le système utilise lorsque le `service` spécifie un protocole d'application sans modificateur de port.
`and-ports`	Le système applique la règle si l'application de paquets est connue et correspond à l'application de règle, et si le port de paquets correspond aux ports dans l'en-tête de règle. Vous ne pouvez pas utiliser `and-ports` dans une règle qui ne précise pas d'application.
`or-ports`	Le système applique la règle si l'une des conditions suivantes est remplie : • L'application du paquet est connue et correspond à l'application de la règle. • L'application de paquets est inconnue et le port de paquets correspond aux ports de la règle. • L'application de paquets ne correspond pas à l'application de règle et les ports de paquets correspondent aux ports de règle. • La règle ne précise pas d'application et les ports de paquets correspondent aux ports de la règle.

Tenez compte des points suivants :

- Vous devez inclure un argument d'application de `service` avec les arguments `service and-ports`.
- Si une règle spécifie plusieurs valeurs dans le tableau ci-dessus, le système applique la dernière valeur apparaissant dans la règle.
- Les arguments de port et d'application peuvent être dans n'importe quel ordre.

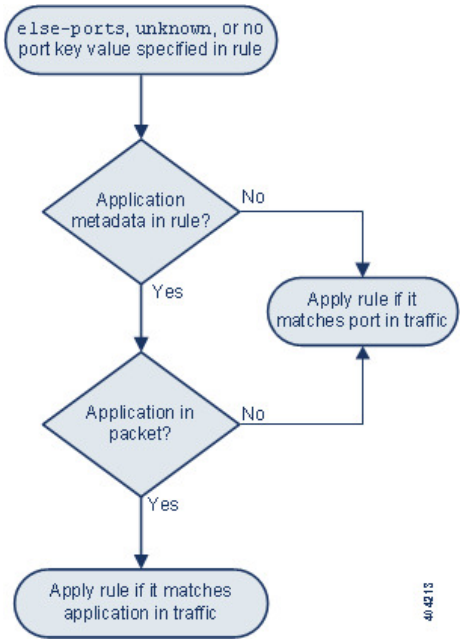
À l'exception de la valeur `and-ports`, vous pouvez inclure un arguments de port de `service` avec ou sans un ou plusieurs arguments d'application `service`. Par exemple :

```
service or-ports, service http, service smtp
```

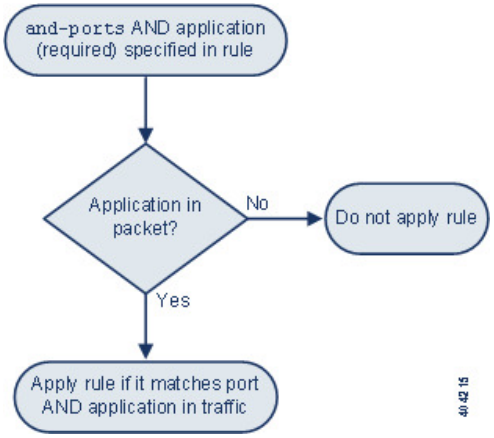
Applications et ports dans le trafic

Les diagrammes ci-dessous illustrent les combinaisons d'application et port prises en charge par les règles de prévention des intrusions, et les résultats de l'application de ces contraintes de règles aux paquets de données.

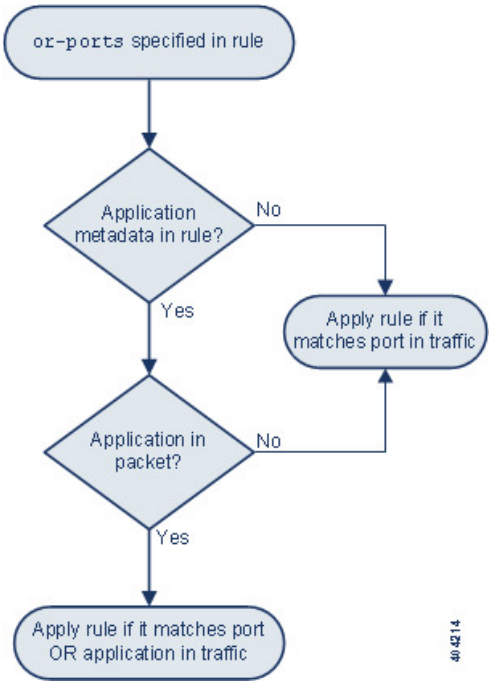
Protocole d'application de l'hôte autres ports source/destination :



Protocole d’application hôte et ports source/destination :



Protocole d'application hôte ou ports source/destination :



Exemples de correspondances

Les exemples de règles suivants utilisant le mot-clé `métadonnées` avec des arguments de `service` sont affichés avec des exemples de données auxquelles ils correspondent et ne correspondent pas :

- `alert tcp any any -> any [80,8080] (metadata:service and-ports, service http, service smtp;)`

Exemples de correspondances	Exemple de non-correspondances
• Trafic HTTP sur le port TCP 80 • Trafic HTTP sur le port TCP 8080 • Trafic SMTP sur le port TCP 80 • Trafic SMTP sur le port TCP 8080	• Trafic POP3 sur les ports 80 ou 8080 • Trafic d'une application inconnue sur les ports 80 ou 8080 • Trafic HTTP sur le port 9999

- `alert tcp any any -> any [80,8080] (metadata:service or-ports, service http;)`

Exemples de correspondances	Exemple de non-correspondances
• Trafic HTTP sur n'importe quel port • Trafic SMTP sur le port 80 • Trafic SMTP sur le port 8080 • Trafic d'application inconnue sur les ports 80 et 8080	• Trafic non HTTP et non SMTP sur les ports autres que le port 80 ou 8080

- L'une des règles suivantes :
  - `alert tcp any any -> any [80,8080] metadata:service else-ports, service http;)`
  - `alert tcp any any -> any [80,8080] metadata:service unknown, service http;)`
  - `alert tcp any any -> any [80,8080] metadata:service http;)`

Exemples de correspondances	Exemple de non-correspondances
• Trafic HTTP sur n'importe quel port • port 80 si l'application de paquet est inconnue • port 8080, si l'application de paquet est inconnue	• Trafic SMTP sur les ports 80 ou 8080 • Trafic POP3 sur les ports 80 ou 8080

Lignes directrices de recherche de métadonnées

Pour rechercher des règles qui utilisent le mot-clé `métadata`, sélectionnez le mot-clé `métadata` sur la page de recherche de règles et, éventuellement, saisissez une partie des métadonnées. Par exemple, vous pouvez taper :

- `search` pour afficher toutes les règles pour lesquelles vous avez utilisé `la recherche` de la *clé*.
- `search http` pour afficher toutes les règles dans lesquelles vous avez utilisé `la recherche` de *clé* et `http` comme *valeur*.
- `author snortguru` pour afficher toutes les règles dans lesquelles vous avez utilisé `author` pour la *clé* et `SnortGuru` comme *valeur*.
- `author s` pour afficher toutes les règles dans lesquelles vous avez utilisé `author` pour la *clé* et des termes comme `SnortGuru` ou `SnortUser1` ou `SnortUser2` comme *valeur*.

Astuces Lorsque vous recherchez à la fois *clé* et *valeur* , utilisez le même opérateur de connexion (égal à [=] ou un espace) dans les recherches que celui utilisé dans l'argument *key value* de la règle; Les recherches effectuées renvoient des résultats différents selon que vous faites suivre *key* d'un égal (=) ou d'un espace.

Notez que quel que soit le format que vous utilisez pour ajouter des métadonnées, le système interprète votre terme de recherche de métadonnées comme tout ou partie d'une *valeur de clé* ou comme *clé=valeur* d'argument. Par exemple, les éléments suivants sont des métadonnées valides qui ne suivent pas un format *valeur de clé* ou *clé=valeur* :

```
ab cd ef gh
```

Cependant, le système interprétera chaque espace dans l'exemple comme un séparateur entre une *clé* et une *valeur* . Ainsi, vous pourriez trouver une règle contenant les exemples de métadonnées en utilisant l'une des recherches suivantes de termes juxtaposés et uniques :

```
cd ef
ef gh
ef
```

mais vous ne localiseriez pas la règle en utilisant la recherche suivante, que le système interpréterait comme un *paramètre de valeur de clé* unique :

```
ab ef
```

Sujets connexes

Recherche de règles, à la page 19

Valeurs d'en-tête IP

Vous pouvez utiliser des mots-clés pour identifier d'éventuelles attaques ou violations de la politique de sécurité dans les en-têtes IP des paquets.

fragbits

Le mot-clé `fragbits` inspecte le fragment et les bits réservés dans l'en-tête IP. Vous pouvez vérifier chaque paquet pour le bit réservé, le bit Plus de fragments et le bit Ne pas fragmenter dans n'importe quelle combinaison.

Tableau 36 : Valeurs des arguments Fragbits

Argument	Description
R	Bit réservé
L	Bit Plus de fragments
D	Bit Ne pas fragmenter

Pour affiner davantage une règle à l'aide du mot-clé `fragbits`, vous pouvez spécifier n'importe quel opérateur décrit dans le tableau suivant après la valeur de l'argument dans la règle.

Tableau 37 : Opérateurs Fragbit

Opérateur	Description
signe plus (**+**)	Le paquet doit correspondre à tous les bits spécifiés.
astérisque (*****)	Le paquet peut correspondre à n'importe lequel des bits spécifiés.
point d'exclamation (**!**)	Le paquet répond aux critères si aucun des bits spécifiés n'est activé.

Par exemple, pour générer un événement pour des paquets dont le bit réservé est activé (et éventuellement d'autres bits), utilisez `R+` comme valeur `fragbits`.

ID

Le mot-clé `id` teste le champ d'identification de fragment d'en-tête IP par rapport à la valeur que vous spécifiez dans l'argument du mot-clé. Certains outils et analyseurs de déni de service définissent ce champ à un numéro spécifique qui est facile à détecter. Par exemple, dans SID 630, qui détecte un balayage de ports Synscan, la valeur `id` est `39426`, la valeur statique utilisée comme numéro d'ID dans les paquets transmis par l'analyseur.

Remarque les valeurs des arguments de l'`id` doivent être numériques.

ipopts

Le mot-clé `IPopts` vous permet de rechercher des paquets pour des options d'en-tête IP spécifiées. Le tableau suivant répertorie les valeurs d'arguments disponibles.

Tableau 38 : Arguments IPoption

Argument	Description
rr (taux de renouvellement)	enregistrer un routage
eol	Fin de la liste
nop	aucune opération
Services techniques	horodatage
sec	Option de sécurité IP
lsrr	routage à source souple
ssrr	routage à source stricte
satid	identifiant de flux

Les analystes surveillent le plus souvent un routage source strict et souple, car ces options peuvent être le signe d'une adresse IP source usurpée.

ip_proto

Le mot-clé `ip_proto` vous permet d'identifier les paquets avec le protocole IP spécifié comme valeur de mot-clé. Vous pouvez spécifier les protocoles IP sous la forme d'un nombre, de 0 à 255. Vous pouvez combiner ces nombres avec les opérateurs suivants : `<`, `>`, ou `!`. Par exemple, pour inspecter le trafic avec un protocole autre que ICMP, utilisez `!1` comme valeur pour le mot-clé `ip_proto`. Vous pouvez également utiliser le mot-clé `ip_proto` plusieurs fois dans une seule règle; notez, cependant, que le moteur de règles interprète plusieurs instances du mot-clé comme ayant une relation boolienne AND. Par exemple, si vous créez une règle contenant `ip_proto:!3; ip_proto:!6`, la règle ignore le trafic utilisant le protocole GGP ET le protocole TCP.

Type de service (tos)

Certains réseaux utilisent la valeur du type de service (ToS) pour établir la priorité pour les paquets circulant sur ce réseau. Le mot-clé `tos` vous permet de tester la valeur ToS de l'en-tête IP du paquet par rapport à la valeur que vous avez spécifiée en tant qu'argument du mot-clé. Les règles utilisant le mot-clé `tos` se déclencheront sur les paquets dont les conditions de service sont définies sur la valeur spécifiée et qui répondent aux autres critères définis dans les règles.

Remarque Les valeurs d'argument pour les `tos` doivent être numériques.

Le champ ToS a été obsolète dans le protocole d'en-tête IP et remplacé par le champ DSCP (Differentiated Services Code Point).

ttl

La valeur de la durée de vie (ttl) d'un paquet indique combien de sauts il peut effectuer avant d'être abandonné. Vous pouvez utiliser le mot-clé `ttl` pour tester la valeur ttl de l'en-tête IP du paquet par rapport à la valeur, ou à la plage de valeurs, que vous spécifiez en tant qu'argument du mot-clé. Il peut être utile de définir le paramètre du mot-clé `ttl` sur une valeur faible telle que 0 ou 1, car les valeurs de durée de vie faibles indiquent parfois un Traceroute ou une tentative d'évitement de prévention des intrusions. (Notez, cependant, que la valeur appropriée pour ce mot-clé dépend du positionnement de votre périphérique géré et de la topologie de votre réseau.) Utilisez la syntaxe suivante :

- Utilisez un entier compris entre 0 et 255 pour définir une valeur spécifique pour la valeur TTL. Vous pouvez également faire précéder la valeur du signe égal (=) (par exemple, vous pouvez spécifier `5` ou `=5`).
- Utilisez un tiret (`-`) pour spécifier une plage de valeurs TTL (par exemple, `0-2` spécifie toutes les valeurs de 0 à 2, `-5` toutes les valeurs de 0 à 5 et `5-`, toutes les valeurs de 5 à 255).
- Utilisez le signe supérieur à (>) pour spécifier des valeurs TTL supérieures à une valeur spécifique (par exemple, `>3` spécifie toutes les valeurs supérieures à 3).
- Utilisez les signes supérieur à et égal à (>=) pour spécifier les valeurs TTL supérieures ou égales à une valeur spécifique (par exemple, `>=3` spécifie toutes les valeurs supérieures ou égales à 3).
- Utilisez le signe inférieur à (<) pour spécifier des valeurs TTL inférieures à une valeur spécifique (par exemple, `<3` spécifie toutes les valeurs inférieures à 3).
- Utilisez les signes inférieur à et égal à (<=) pour spécifier des valeurs TTL inférieures ou égales à une valeur spécifique (par exemple, `<=3` spécifie toutes les valeurs inférieures ou égales à 3).

Valeurs d'en-tête ICMP

Le système Firepower prend en charge les mots-clés que vous pouvez utiliser pour identifier les attaques et les violations de la politique de sécurité dans les en-têtes des paquets ICMP. Notez, cependant, qu'il existe des règles prédéfinies qui détectent la plupart des types et des codes ICMP. Envisagez d'activer une règle existante ou de créer une règle locale basée sur une règle existante; vous pourrez peut-être trouver une règle qui répond à vos besoins plus rapidement que si vous élaboriez une règle ICMP de toutes pièces.

icmp_id and icmp_seq

L'identification ICMP et les numéros de séquence permettent d'associer les réponses ICMP aux requêtes ICMP. Dans le trafic normal, ces valeurs sont affectées dynamiquement aux paquets. Certains programmes de canal secret et de déni de serveur distribué (DDoS) utilisent un ID ICMP statique et des valeurs de séquence. Les mots-clés suivants vous permettent d'identifier les paquets ICMP avec des valeurs statiques.

Mot-clé	Définition
icmp_id	Inspecte le numéro d'ID ICMP d'une demande ECHO ICMP ou d'un paquet de réponse. Utilisez une valeur numérique qui correspond au numéro d'ID ICMP comme arguments du mot-clé `icmp_id`.
icmp_seq	Le mot-clé `icmp_seq` inspecte la séquence ICMP d'une requête ECHO ICMP ou d'un paquet de réponse. Utilisez une valeur numérique qui correspond au numéro de séquence ICMP comme arguments du mot-clé `icmp_seq`.

itype

Utilisez le mot-clé `itype` pour rechercher les paquets avec des valeurs de type de message ICMP spécifiques. Vous pouvez spécifier une valeur de type ICMP valide ou non valide pour tester les différents types de trafic. Par exemple, les attaquants peuvent définir les valeurs de type ICMP hors des limites pour provoquer des attaques par déni de service et par flooding.

Vous pouvez spécifier une plage pour la valeur de l'argument `itype` en utilisant inférieur à (|) et supérieur à (>).

Par exemple :

- <35
- >36
- 3<>55

icode

Les messages ICMP comprennent parfois une valeur de code qui fournit des détails lorsqu'une destination est inaccessible.

Vous pouvez utiliser le mot-clé `icode` pour identifier les paquets avec des valeurs de code ICMP spécifiques. Vous pouvez choisir de spécifier une valeur de code ICMP valide ou non valide pour tester les différents types de trafic.

Vous pouvez spécifier une plage pour la valeur d'argument `icode` en utilisant moins de (|) et plus de (>).

Par exemple :

- pour trouver des valeurs inférieures à 35, spécifiez `<35`.
- pour trouver les valeurs supérieures à 36, spécifiez `>36`.
- pour trouver des valeurs comprises entre 3 et 55, spécifiez `3<>55`.

Astuces Vous pouvez utiliser les mots-clés `icode` et `itype` ensemble pour identifier le trafic qui correspond aux deux. Par exemple, pour identifier le trafic ICMP qui contient un type de code ICMP Destination Unreachable avec un type de code ICMP Port Unreachable, spécifiez un mot-clé `itype` avec une valeur de 3 (pour Destination Unreachable) et un mot-clé `icode` avec une valeur de 3 (pour Port Unreachable).

Valeurs d'en-tête TCP et taille du flux

Le système Firepower prend en charge les mots-clés conçus pour identifier les attaques tentées à l'aide des en-têtes TCP des paquets et de la taille des flux TCP.

ack

Vous pouvez utiliser le mot-clé `ack` pour comparer une valeur au numéro d'accusé de réception TCP d'un paquet. La règle se déclenche si le numéro d'accusé de réception TCP d'un paquet correspond à la valeur spécifiée pour le mot-clé `ack`.

Les valeurs d'argument de `ack` doivent être numériques.

flags

Vous pouvez utiliser le mot-clé `flags` pour spécifier n'importe quelle combinaison d'indicateurs TCP qui, lorsqu'ils sont définis dans un paquet inspecté, entraînent le déclenchement de la règle.

Remarque Dans les situations où vous utilisiez traditionnellement `A+` comme valeur pour `flags`, vous devez plutôt utiliser le mot-clé `flow` avec la valeur `established`(établi). En général, vous devez utiliser le mot-clé `flow` avec une valeur `stateless` lorsque vous utilisez des indicateurs pour vous assurer que toutes les combinaisons d'indicateurs sont détectées.

Vous pouvez vérifier ou ignorer les valeurs décrites dans le tableau suivant pour le mot-clé `flag`.

Tableau 39 : Arguments de l'indicateur

Argument	TCP Flag (indicateur TCP)
AR	Reconnaît les données
Psh	Les données doivent être envoyées dans ce paquet
Syn	Une nouvelle connexion.
Urg	Le paquet contient des données urgentes
Fin	Une connexion fermée
Rst	Une connexion interrompue
CWR	Une fenêtre de congestion ECN a été réduite C'était auparavant l'argument R1, qui est toujours pris en charge pour la compatibilité ascendante.
ECE	Écho ECN C'était auparavant l'argument R2, qui est toujours pris en charge pour la compatibilité ascendante.

Lorsque vous utilisez le mot-clé `flags`, vous pouvez utiliser un opérateur pour indiquer comment le système effectue les correspondances avec plusieurs indicateurs. Le tableau suivant décrit ces options.

Tableau 40 : Opérateurs utilisés avec les indicateurs

Opérateur	Description	Exemple
tous	Le paquet doit contenir tous les indicateurs spécifiés.	Sélectionnez `Urg` et `all` pour préciser qu'un paquet doit contenir l'indicateur Urgent et peut contenir tout autre indicateur.
Tous	Le paquet contient tous, quelques-uns ou aucun des indicateurs spécifiés.	Sélectionnez `Ack`, `Psh` et `any` pour préciser que l'un des indicateurs `Ack` et `Psh`, ou les deux, doit être défini pour déclencher l'application de la règle, et que d'autres indicateurs peuvent également être définis sur un paquet.
pas	Le paquet ne doit **pas** contenir l'ensemble d'indicateurs spécifié.	Sélectionnez `Urg` et `not` pour préciser que l'indicateur Urgent n'est **pas** défini pour les paquets qui déclenchent cette règle.

flux

Vous pouvez utiliser le mot-clé `flow` pour sélectionner les paquets à inspecter par une règle en fonction des caractéristiques de la session. Le mot-clé `flow` vous permet de préciser la direction du flux de trafic auquel une règle s'applique, en appliquant les règles au flux client ou au flux serveur. Pour préciser comment le mot-clé `flow` inspecte vos paquets, vous pouvez définir la direction du trafic que vous souhaitez analyser, l'état des paquets inspectés et si les paquets font partie d'un flux recréé.

L'inspection dynamique des paquets se produit lors du traitement des règles. Si vous souhaitez qu'une règle TCP ignore le trafic sans état (trafic sans contexte de session établi), vous devez ajouter le mot-clé `flow` à la règle et sélectionner l'argument **Established** pour le mot-clé. Si vous souhaitez qu'une règle UDP ignore le trafic sans état, vous devez ajouter le mot-clé `flow` à la règle et sélectionner l'argument **Established** ou un argument directionnel, ou les deux. Ainsi, la règle TCP ou UDP effectue une inspection dynamique d'un paquet.

Lorsque vous ajoutez un arguments directionnels, le moteur de règles inspecte uniquement les paquets qui ont un état établi avec un flux qui correspond à la direction spécifiée. Par exemple, si vous ajoutez le mot-clé `flow` avec l'argument `established` et l'argument `from Client` à une règle qui se déclenche lorsqu'une connexion TCP ou UDP est détectée, le moteur de règles inspecte uniquement les paquets envoyés par le client.

Astuces Pour des performances maximales, incluez toujours un mot-clé de `flow` dans une règle TCP ou une règle de session UDP.

Le tableau suivant décrit les arguments liés au flux que vous pouvez spécifier pour le mot-clé `flow` :

Tableau 41 : Arguments de flux liés à l'état

Argument	Description
Établi	Se déclenche sur les connexions établies.
Sans état	Se déclenche quel que soit l'état du processeur de flux.

Le tableau suivant décrit les options de direction que vous pouvez spécifier pour le mot-clé `flow` :

Tableau 42 : Arguments directionnels du flux

Argument	Description
Au client	Déclencheurs sur les réponses du serveur.
Vers le serveur	Déclencheurs sur réponses des clients.
Du client	Déclencheurs sur réponses des clients.
À partir du serveur	Déclencheurs sur les réponses du serveur.

Vous constaterez que `From Server` et `to Client` remplissent la même fonction, tout comme `From Server` et `From Client`. Ces options existent pour ajouter du contexte et de la lisibilité à la règle. Par exemple, si vous créez une règle conçue pour détecter une attaque d'un serveur vers un client, utilisez `From server`. En revanche, si vous créez une règle conçue pour détecter une attaque du client vers le serveur, utilisez l'option `From Client`.

Le tableau suivant décrit les arguments liés au flux que vous pouvez spécifier pour le mot-clé `flow` :

Tableau 43 : Arguments de flux lié au flux

Argument	Description
Ignorer le trafic de flux	Ne se déclenche pas sur les paquets de flux recréés.
Flux de trafic uniquement	Se déclenche uniquement sur les paquets de flux recréés.

Par exemple, vous pouvez utiliser `To Server, Existing, Only Stream Traffic` comme valeur pour le mot-clé `flow` afin de détecter le trafic, circulant d'un client au serveur dans une session établie, et qui a été réassemblé par le préprocesseur de flux.

seq

Le mot-clé `seq` vous permet de spécifier une valeur de numéro de séquence statique. Les paquets dont le numéro de séquence correspond à l'argument spécifié déclenche la règle contenant le mot-clé. Bien que ce mot-clé soit rarement utilisé, il est utile pour identifier les attaques et les analyses de réseau qui utilisent des paquets générés avec des numéros de séquence statiques.

window

Vous pouvez utiliser le mot-clé `window` pour préciser la taille de la fenêtre TCP qui vous intéressent. Une règle contenant ce mot-clé se déclenche chaque fois qu'elle rencontre un paquet avec la taille de fenêtre TCP spécifiée. Bien que ce mot-clé soit rarement utilisé, il est utile pour identifier les attaques et les analyses de réseau qui utilisent des paquets générés avec des tailles de fenêtre TCP statiques.

stream_size

Vous pouvez utiliser le mot-clé `stream_size` conjointement avec le préprocesseur de flux pour déterminer la taille en octets d'un flux TCP, en utilisant le format :

```
direction,operator,bytes
```

où octets est le nombre d'octets. Vous devez séparer chaque option de l'argument par une virgule (,).

Le tableau suivant décrit les options directionnelles non sensibles à la casse que vous pouvez spécifier pour le mot-clé `stream_size` :

Tableau 44 : Arguments directionnels du mot-clé stream_size

Argument	Description
client	se déclenche sur un flux du client correspondant à la taille de flux spécifiée.
serveur	se déclenche sur un flux du serveur correspondant à la taille de flux spécifiée.
les deux	se déclenche sur le trafic du client et du serveur correspondant tous deux à la taille de flux spécifiée. Par exemple, l'argument les `both, >, 200` se déclencherait lorsque le trafic du client est supérieur à 200 octets ET que le trafic du serveur est supérieur à 200 octets.
either	se déclenche sur le trafic du client ou du serveur correspondant à la taille de flux spécifiée, selon la première éventualité. Par exemple, l'argument `, >, 200` se déclencherait lorsque le trafic du client est supérieur à 200 octets OU que le trafic du serveur est supérieur à 200 octets.

Le tableau suivant décrit les opérateurs que vous pouvez utiliser avec le mot-clé `stream_size` :

Tableau 45 : Opérateurs d'arguments de mot-clé stream_size

Opérateur	Description
=	égal à
!=	Différent de
>	supérieur à
<	inférieur à
>=	supérieur ou égal à
<=	est inférieur ou égal à

Par exemple, vous pouvez utiliser `client, >=, 5001216` comme paramètre du mot-clé `stream_size` afin de détecter un flux TCP circulant d'un client à un serveur et supérieur ou égal à 5001216 octets.

Le mot-clé stream_reassembly

Vous pouvez utiliser le mot-clé `stream_reassemble` pour activer ou désactiver le réassemblage du flux TCP pour une seule connexion lorsque le trafic inspecté sur la connexion correspond aux conditions de la règle. Vous pouvez également utiliser ce mot-clé plusieurs fois dans une règle.

Utilisez la syntaxe suivante pour activer ou désactiver le réassemblage du flux :

```
enable|disable, server|client|both, option, option
```

Le tableau suivant décrit les arguments facultatifs que vous pouvez utiliser avec le mot-clé `stream_reassemble`.

Tableau 46 : Arguments facultatifs flux_reassemble

Argument	Description
pas d'alerte	Ne génère aucun événement, quelles que soient les autres options de détection spécifiées dans la règle.
fastpath	Ignorez le reste du trafic de connexion lorsqu'il y a une correspondance.

Par exemple, la règle suivante désactive le réassemblage du flux TCP côté client sans générer d'événement sur la connexion, où un code d'état 200 OK est détecté dans une réponse HTTP :

```
alert tcp any 80 -> any any (flow:to_client, established; content: "200 OK";
stream_reassemble:disable, client, noalert
```

Mots-clés SSL

Vous pouvez utiliser des mots-clés de règles SSL pour appeler le préprocesseur du protocole SSL (Secure sockets Layer) et extraire des informations sur la version SSL et l'état de la session des paquets d'une session chiffrée.

Lorsqu'un client et un serveur communiquent pour établir une session chiffrée à l'aide de SSL ou de Transport Layer Security (TLS), ils échangent des messages d'établissement de liaison. Bien que les données transmises au cours de la session soient chiffrées, les messages d'établissement de liaison ne le sont pas.

Le préprocesseur SSL extrait les informations d'état et de version de champs d'établissement de liaison spécifiques. Deux champs dans l'établissement de liaison indiquent la version de SSL ou TLS utilisée pour chiffrer la session et l'étape de l'établissement de liaison.

ssl_state

Le mot-clé `ssl_state` peut être utilisé pour la mise en correspondance avec les informations d'état pour une session chiffrée. Pour vérifier deux versions SSL ou plus utilisées simultanément, utilisez plusieurs mots-clés `ssl_version` dans une règle.

Lorsqu'une règle utilise le mot-clé `ssl_state`, le moteur de règles fait appel au préprocesseur SSL pour vérifier le trafic à la recherche d'informations sur l'état SSL.

Par exemple, pour détecter la tentative d'un attaquant de provoquer un débordement de la mémoire tampon sur un serveur en envoyant un message `ClientHello` avec une longueur de défi trop longue et trop de données, vous pouvez utiliser le mot-clé `ssl_state` avec `client_hello` comme argument, puis vérifier les paquets anormalement volumineux.

Utilisez une liste séparée par des virgules pour spécifier plusieurs arguments pour l'état SSL. Lorsque vous dressez la liste de plusieurs arguments, le système les évalue à l'aide de l'opérateur OU. Par exemple, si vous spécifiez `client_hello` et `server_hello` comme arguments, le système évalue la règle par rapport au trafic qui comporte un `client_hello` OU un `server_hello`.

Vous pouvez également annuler n'importe quel arguments. Par exemple :

```
!client_hello, !unknown
```

Pour s'assurer que la connexion a atteint chacun d'un ensemble d'états, plusieurs règles utilisant l'option de règle ssl_state doivent être utilisées. Le mot-clé `ssl_state` accepte les identifiants suivants comme arguments :

Tableau 47 : Arguments `ssl_state`

Argument	Objectif
`client_hello`	Correspondance avec un message d'établissement de liaison avec `ClientHello` comme type de message, où le client demande une session chiffrée.
`server_hello`	Correspondance avec un message d'établissement de liaison avec `ServerHello` comme type de message, dans lequel le serveur répond à la demande du client d'ouvrir une session chiffrée.
`client_keyx`	Correspondance avec un message d'établissement de liaison avec `ClientKeyExchange` comme type de message, dans lequel le client transmet une clé au serveur pour confirmer la réception d'une clé du serveur.
`server_keyx`	Correspondance avec un message d'établissement de liaison avec `ServerKeyExchange` comme type de message, dans lequel le client transmet une clé au serveur pour confirmer la réception d'une clé du serveur.
`inconnu`	Correspondances avec n'importe quel type de message d'établissement de liaison.

ssl_version

Le mot-clé `ssl_version` peut être utilisé pour la mise en correspondance avec les informations de version pour une session chiffrée. Lorsqu'une règle utilise le mot-clé `ssl_version`, le moteur de règles fait appel au préprocesseur SSL pour vérifier le trafic des informations sur la version SSL.

Par exemple, si vous savez qu'il existe une vulnérabilité de débordement de tampon dans SSL version 2, vous pourriez utiliser le mot-clé `ssl_version` avec l'argument `sslv2` pour identifier le trafic utilisant cette version de SSL.

Utilisez une liste séparée par des virgules pour spécifier plusieurs arguments pour la version SSL. Lorsque vous dressez la liste de plusieurs arguments, le système les évalue à l'aide de l'opérateur OU. Par exemple, si vous souhaitez identifier le trafic chiffré qui n'utilise pas SSLv2, vous pouvez ajouter `ssl_version:ssl_v3,tls1.0,tls1.1,tls1.2` à une règle. La règle évaluerait tout trafic à l'aide de SSL version 3, TLS version 1.0, TLS version 1.1 ou TLS version 1.2.

Le mot-clé `ssl_version` accepte les identifiants de version SSL/TLS suivants comme arguments :

Tableau 48 : Arguments `ssl_version`

Argument	Objectif
`sslv2`	Correspondance avec le trafic codé à l'aide du protocole SSL (Secure socket Layer) version 2.
`sslv3`	Correspondance avec le trafic codé à l'aide du protocole SSL (Secure socket Layer) version 3.
`tls1.0`	Correspondance avec le trafic codé à l'aide de TLS (Transport Layer Security) version 1.0.
`tls1.1`	Correspondance avec le trafic codé à l'aide de Transport Layer Security (TLS) version 1.1.

Argument	Objectif
`tls1.2`	Correspondance avec le trafic codé à l'aide de Transport Layer Security (TLS) version 1.2.

Le mot-clé appid

Vous pouvez utiliser le mot-clé `appid` pour identifier le protocole d'application, l'application client ou l'application Web dans un paquet. Par exemple, vous pouvez cibler une application particulière que vous savez sensible à une vulnérabilité particulière.

Dans le mot-clé `appid` d'une règle de prévention des intrusions, cliquez sur **Configure AppID** (Configurer AppID) pour sélectionner une ou plusieurs applications que vous souhaitez détecter.

Parcourir les applications disponibles

Lorsque vous commencez à créer la condition, la liste des **applications disponibles** n'est pas limitée et affiche toutes les application détectées par le système, à raison de 100 par page :

- Pour faire défiler les applications, cliquez sur les flèches sous la liste.
- Pour afficher une fenêtre contextuelle contenant des renseignements sommaires sur les caractéristiques de l'application, ainsi que des liens de recherche Internet que vous pouvez suivre, cliquez sur **Information** () à côté d'une application.

Utilisation des filtres d'application

Pour vous aider à trouver les applications que vous souhaitez mettre en correspondance, vous pouvez limiter la liste des **applications disponibles** comme suit :

- Pour rechercher des applications, cliquez sur le bouton **Rechercher par nom** au-dessus de la liste, puis saisissez un nom. La liste est mise à jour à mesure que vous saisissez pour afficher les applications correspondantes.
- Pour restreindre les applications en appliquant un filtre, utilisez la liste **Filtres d'application**. La liste des **applications disponibles** est mise à jour à mesure que vous appliquez des filtres. Pour votre commodité, le système utilise une **icône de déverrouillage** pour marquer les applications que le système peut identifier uniquement dans le trafic déchiffré, et non pas chiffrés ou non chiffrés.

Remarque Si vous sélectionnez un ou plusieurs filtres dans la liste Filtres d'applications et que vous effectuez également une recherche dans la liste des **applications disponibles**, vos sélections et la liste des **applications disponibles** filtrée par la recherche sont combinées à l'aide d'une opération AND.

Sélection des applications

Pour ne sélectionner qu'une seule application, sélectionnez-la et cliquez sur **Add to Rule** (Ajouter à la règle). Pour sélectionner toutes les applications dans la vue sous filtrage actuel, cliquez avec le bouton droit et sélectionnez **Sélectionner tout**.

Valeurs du protocole de la couche applicative

Bien que les préprocesseurs effectuent la plupart de la normalisation et de l'inspection des valeurs de protocole de la couche d'application, vous pouvez continuer à inspecter les valeurs de la couche d'application en utilisant diverses options de préprocesseur.

Le mot-clé RPC

Le mot-clé `rpc` identifie les services d'appel de procédure à distance ONC (Open Network Computing Remote Procédure Call) dans les paquets TCP ou UDP. Cela vous permet de détecter les tentatives d'identification des programmes RPC sur un hôte. Les intrus peuvent utiliser un mappeur de port RPC pour déterminer si l'un des services RPC en cours d'exécution sur votre réseau peut être exploité. Ils peuvent également tenter d'accéder à d'autres ports exécutant l'appel RPC sans utiliser de mappeur de port. Le tableau suivant répertorie les arguments acceptés par le mot-clé `rpc`.

Tableau 49 : Arguments du mot-clé rpc

Argument	Description
application	Le numéro d'application d'appel RPC
procedure	La procédure RPC appelée
version	La version d'appel RPC

Pour définir les arguments du mot-clé `rpc`, utilisez la syntaxe suivante :

```
application,procedure,version
```

où application est le numéro de l'application RPC, procédure est le numéro de procédure RPC et version est le numéro de version RPC. Vous devez préciser tous les arguments du mot-clé `rpc`. Si vous n'êtes pas en mesure de préciser l'un des arguments, remplacez-le par un astérisque (`*`).

Par exemple, pour rechercher le mappeur de port RPC (qui est l'application RPC indiquée par le nombre 100000), avec n'importe quelle procédure ou version, utilisez `100000,*,*` comme arguments.

Le mot-clé asn.1

Le mot-clé `asn.1` vous permet de décoder un paquet ou une partie d'un paquet, à la recherche de divers encodages malveillants.

Le tableau suivant décrit les arguments du mot-clé `asn.1`.

Tableau 50 : Arguments du mot clé asn.1

Argument	Description
Débordement de la chaîne de bits	Détecte les encodages de chaînes de bits non valides connus pour être exploitables à distance
Double débordement	Détecte le double encodage ascii supérieur à une mémoire tampon standard On sait qu'il s'agit d'une fonction exploitable de Microsoft Windows, mais on ne sait pas quels services pourraient être exploités.

Argument	Description
Longueur surdimensionnée	Détecte les longueurs de type ASN.1 supérieures à l'argument fourni. Par exemple, si vous définissez la longueur surdimensionnée sur 500, tout type ASN.1 supérieur à 500 déclenche la règle.
Décalage absolu	Définit un décalage absolu à partir du début de la charge utile du paquet. (Rappelez-vous que le compteur de décalage commence à l'octet 0.) Par exemple, si vous souhaitez décoder des paquets SNMP, définissez le décalage absolu sur 0 et ne définissez pas de décalage relatif. le décalage absolu peut être positif ou négatif.
Décalage relatif	Il s'agit du décalage relatif à partir de la dernière correspondance de contenu réussie, `PCre`ou `byte_jump`. Pour décoder une séquence ASN.1 juste après le contenu « foo », définissez le décalage relatif sur 0 et ne définissez pas de décalage absolu. Le décalage relatif peut être positif ou négatif. (Rappelez-vous que le compteur de décalage commence à 0.)

Par exemple, il existe une vulnérabilité connue dans la bibliothèque ASN.1 de Microsoft qui entraîne un débordement de la mémoire tampon, permettant à un attaquant d'exploiter la condition avec un paquet d'authentification spécialement conçu. Lorsque le système décode les données au format asn.1, le code d'exploitation du paquet pourrait s'exécuter sur l'hôte avec des privilèges de niveau système ou pourrait provoquer une situation de déni de service. La règle suivante utilise le mot-clé `asn1` pour détecter les tentatives d'exploitation de cette vulnérabilité :

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(flow:to_server, established; content:"|FF|SMB|73|";
nocase; offset:4; depth:5;
asn1:bitstring_overflow,double_overflow,oversize_length 100,
relative_offset 54;)
```

La règle ci-dessus génère un événement par rapport au trafic TCP circulant de n'importe quelle adresse IP définie dans la variable $EXTERNAL_NET, depuis n'importe quel port, vers n'importe quelle adresse IP définie dans la variable $Home_NET en utilisant le port 445. En outre, il exécute la règle uniquement sur les connexions TCP aux serveurs établies. La règle teste ensuite le contenu spécifique dans des emplacements spécifiques. Enfin, la règle utilise le mot-clé `asn1` pour détecter les encodages de chaînes de bits et les encodages ASCII doubles et pour identifier les longueurs de type asn.1 supérieures à 100 octets en commençant à 55 octets après la fin de la dernière correspondance de contenu réussie. (Rappelez-vous que le compteur de `décalage` commence à l'octet 0.)

Le mot-clé urilen

Vous pouvez utiliser le mot-clé `urilen` conjointement avec le préprocesseur HTTP Inspect pour inspecter le trafic HTTP à la recherche d'URI d'une longueur spécifique, inférieure à la longueur maximale, supérieure à une longueur minimale ou dans une plage spécifiée.

Une fois que le préprocesseur HTTP Inspect s'est normalisé et a inspecté le paquet, le moteur de règles évalue le paquet par rapport à la règle et détermine si l'URI correspond à la condition de longueur spécifiée par le mot-clé `urilen`. Vous pouvez utiliser ce mot-clé pour détecter les exploits qui tentent de tirer parti des vulnérabilités de la longueur d'URI, par exemple, en créant un débordement de la mémoire tampon qui permet à l'attaquant de provoquer une condition de DoS ou d'exécuter du code sur l'hôte avec des privilèges de niveau système.

Tenez compte des éléments suivants lorsque vous utilisez le mot-clé `urilen` dans une règle :

- En pratique, vous utilisez toujours le mot-clé `urilen` en combinaison avec le mot-clé `flow.established` et un ou plusieurs autres mots-clés.
- Le protocole de règles est toujours TCP.
- Les ports cibles sont toujours des ports HTTP.

Vous spécifiez la longueur d'URI à l'aide d'un nombre décimal d'octets, inférieur à (<) et supérieur à (>).

Par exemple :

- Spécifiez `5` pour détecter un URI de 5 octets de long.
- Spécifiez `< 5` (séparés par un espace) pour détecter un URI de moins de 5 octets.
- Spécifiez `> 5` (séparés par un caractère d'espace) pour détecter un URI supérieure à 5 octets de long.
- Spécifiez `3 <> 5` (avec un espace avant et après <>pour détecter un URI ayant une longueur de 3 à 5 octets.

Par exemple, il y a une vulnérabilité connue dans la version 2.4 de l'utilitaire de surveillance et de dépistage de serveur de Novell iMonitor, qui accompagne la version 8.8 de eDirectory. Un paquet contenant un URI excessivement long provoque un débordement de la mémoire tampon, ce qui permet à un attaquant d'exploiter la condition avec un paquet spécialement conçu qui pourrait s'exécuter sur l'hôte avec des privilèges de niveau système ou qui pourrait provoquer un problème de déni de service. La règle suivante utilise le mot-clé `urilen` pour détecter les tentatives d'exploitation de cette vulnérabilité :

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS
(msg:"EXPLOIT eDirectory 8.8 Long URI iMonitor buffer
overflow attempt"; flow:to_server,established;
urilen:> 8192; uricontent:"/nds/"; nocase;
classtype:attempted-admin; sid:x; rev:1;)
```

La règle ci-dessus génère un événement par rapport au trafic TCP circulant depuis n'importe quelle adresse IP définie dans la variable $EXTERNAL_NET, depuis n'importe quel port, vers n'importe quelle adresse IP définie dans la variable $HOME_NET en utilisant les ports définis dans la variable $HTTP_PORTS. En outre, les paquets sont évalués par rapport à la règle uniquement sur les connexions TCP aux serveurs établies. La règle utilise le mot-clé `urilen` pour détecter tout URI de plus de 8 192 octets. Enfin, la règle recherche dans l'URI le contenu non sensible à la casse`/nds/`.

Sujets connexes

Protocole d'en-tête de règle de prévention des intrusions, à la page 5
Ports source et de destination de l'en-tête de la règle de prévention des intrusions, à la page 9
Variables prédéfinies par défaut

Mots-clés DCE/RPC

Les trois mots-clés DCE/RPC décrits dans le tableau suivant vous permettent de surveiller les exploits dans le trafic de session DCE/RPC. Lorsque le système traite des règles avec ces mots clés, il appelle le préprocesseur DCE/RPC.

Tableau 51 : Mots-clés DCE/RPC

Utiliser...	De cette façon...	Pour détecter...
`dce_iface`	seul	paquets identifiant un service DCE/RPC précis
`dce_opnum`	précédé de `dce_iface`	paquets identifiant des opérations de service DCE/RPC spécifiques
`dce_stub_data`	précédé de `dce_iface` + `dce_opnum`	données tampons définissant une demande ou une réponse à l'opération précise

Dans le tableau, vous devez toujours faire précéder `dce_opnum` de `dce_iface` et que vous devez toujours faire précéder `dce_stub_data` de `dce_iface` + `dce_opnum`.

Vous pouvez également utiliser ces mots-clés DCE/RPC avec d'autres mots-clés de règles. Notez que pour les règles DCE/RPC, vous utilisez les mots-clés `byte_jump`, `byte_test` et `byte_extract` avec leurs arguments **DCE/RPC** sélectionnés.

Cisco vous recommande d'inclure au moins un mot-clé `content` dans les règles qui comprennent des mots-clés DCE/RPC pour vous assurer que le moteur de règles utilise la correspondance de modèle rapide, ce qui accélère la vitesse de traitement et améliore les performances. Notez que le moteur de règles utilise la correspondance de modèle rapide lorsqu'une règle comprend au moins un mot-clé `content`, que vous ayez ou non activé l'argument **Use Fast Pattern Matcher** (utiliser la correspondance de modèle rapide) pour le mot-clé `content`.

Vous pouvez utiliser la version de DCE/RPC et les informations d'en-tête adjacentes comme contenu correspondant dans les cas suivants :

- la règle ne comprend pas d'autre mot-clé `content`
- la règle contient un autre mot-clé `content`, mais la version DCE/RPC et les informations adjacentes représentent un modèle plus unique que l'autre contenu

  Par exemple, la version DCE/RPC et les informations adjacentes sont plus susceptibles d'être uniques qu'un seul octet de contenu.

Vous devez mettre fin aux règles admissibles avec l'une des versions suivantes et les correspondances de contenu d'information adjacentes :

- Pour les règles DCE/RPC axées sur la connexion, utiliser le contenu `|05 00 00|` (pour la version majeure 05, la version mineure 00 et la demande d'unité de données de protocole (PDU) de type 00).
- Pour les règles DCE/RPC sans connexion, utiliser le contenu `|04 00|` (pour la version 04 et la demande de PDU de type 00).

Dans les deux cas, placez le mot-clé de `content` pour la version et les informations adjacentes comme dernier mot-clé dans la règle pour appeler l'outil de correspondance de modèle rapide sans répéter le traitement déjà terminé par le préprocesseur DCE/RPC. Notez que le fait de placer le mot-clé « `content` » à la fin de la règle s'applique au contenu de la version utilisée en tant que périphérique pour appeler l'outil de correspondance de modèle rapide, et pas nécessairement aux autres correspondances de contenu dans la règle.

Sujets connexes

Le préprocesseur DCE/RPC
Les mots-clés content et protected_content, à la page 25
Arguments de la recherche de schéma rapide pour mot-clé de contenu, à la page 35
Présentation : mots-clés byte_jump et byte_test

Le mot-clé byte_extract, à la page 43

dce_iface

Vous pouvez utiliser le mot-clé `dce_iface` pour identifier un service DCE/RPC spécifique.

Vous pouvez également utiliser `dce_iface` en combinaison avec les mots-clés `dce_opnum` et `dce_stub_data` pour limiter davantage le trafic DCE/RPC à inspecter.

Un identifiant unique universel (UUID) fixe de 16 octets identifie l'interface d'application attribuée à chaque service DCE/RPC. Par exemple, l'UUID 4b324fc8-670-01d3-1278-5a47bf6ee188 identifie le service lanmanserver DCE/RPC, également connu sous le nom de service srvsvc, qui fournit de nombreuses fonctions de gestion pour le partage de périphériques homologues, de fichiers et de canaux nommés SMB. Le préprocesseur DCE/RPC utilise l'UUID et les valeurs d'en-tête associées pour suivre les sessions DCE/RPC.

L'UUID d'interface est composé de cinq chaînes hexadécimales séparées par des tirets :

```
<4hexbytes>-<2hexbytes>-<2hexbytes>-<2hexbytes>-<6hexbytes>
```

Vous spécifiez l'interface en saisissant l'UUID complet, y compris les tirets, comme le montre l'UUID suivant pour l'interface réseau :

```
12345678-1234-abcd-ef00-01234567cffb
```

Notez que vous devez spécifier les trois premières chaînes de l'UUID dans l'ordre des octets big endian. Bien que les listes d'interfaces publiées et les analyseurs de protocole affichent généralement les UUID dans le bon ordre des octets, vous devrez peut-être réorganiser l'ordre des octets de l'UUID avant de le saisir. Considérez l'UUID du service de messagerie suivant, car il peut parfois s'afficher en texte ASCII brut avec les trois premières chaînes dans l'ordre des octets little endian :

```
f8 91 7b 5a 00 ff d0 11 a9 b2 00 c0 4f b6 e6 fc
```

Vous devez spécifier le même UUID pour le mot-clé `dce_iface` en insérant des tirets et en mettant les trois premières chaînes dans l'ordre des octets au format big endian, comme suit :

```
5a7b91f8-ff00-11d0-a9b2-00c04fb6e6fc
```

Bien qu'une session DCE/RPC puisse inclure des demandes vers plusieurs interfaces, vous ne devez inclure qu'un seul mot-clé `dce_iface` dans une règle. Créez des règles supplémentaires pour détecter des interfaces supplémentaires.

Les interfaces d'applications DCE/RPC ont également des numéros de version d'interface. Vous pouvez éventuellement spécifier une version d'interface à l'aide d'un opérateur indiquant que la version est égale, différente de la version, est inférieure ou supérieure à la valeur spécifiée.

L'ETCD/RPC orienté et sans connexion peut être fragmenté en plus de toute segmentation TCP ou IP. En règle générale, il n'est pas utile d'associer un fragment DCE ou RPC autre que le premier à l'interface spécifiée, car cela pourrait entraîner un grand nombre de faux positifs. Cependant, pour des raisons de flexibilité, vous pouvez éventuellement évaluer tous les fragments par rapport à l'interface spécifiée.

Le tableau suivant résume les arguments du mot-clé `dce_iface`.

Tableau 52 : Arguments `dce_iface`

Argument	Description
UUID de l'interface	L'UUID, y compris les tirets, qui identifie l'interface d'application du service spécifique que vous souhaitez détecter dans le trafic DCE/RPC. Toute demande associée à l'interface spécifiée correspondrait à l'UUID de l'interface.
Version	Facultativement, le numéro de version de l'interface de l'application de 0 à 65535 et un opérateur indiquant s'il faut détecter une version supérieure (>), inférieure à (|), égale (=) ou différente de (!) à la valeur spécifiée.
Tous les fragments	Le cas échéant, la mise en correspondance avec l'interface dans tous les fragments DCE/RPC associés et, si spécifié, sur la version de l'interface. Cet arguments est désactivé par défaut, ce qui indique que le mot-clé ne correspond que si le premier fragment ou l'ensemble du paquet non fragmenté est associé à l'interface spécifiée. Notez que l'activation de cet arguments peut entraîner des faux positifs.

Le mot-clé dce_opnum

Vous pouvez utiliser le mot-clé `dce_opnum` conjointement avec le préprocesseur DCE/RPC pour détecter les paquets qui identifient une ou plusieurs opérations spécifiques effectuées par un service DCE/RPC.

Les appels de fonction client demandent des fonctions de service spécifiques, appelées *opérations*. Un numéro d'opération (opnum) identifie une opération précise dans l'en-tête DCE/RPC. Il est probable qu'un exploit cible une opération précise.

Par exemple, l'UUID 12345678-1234-abcd-ef00-01234567cffb identifie l'interface du service netlogon, qui effectue plusieurs dizaines d'opérations différentes. L'une d'elles est l'opération 6, l'opération NetrServerPasswordSet.

Vous devez faire précéder le mot-clé `dce_opnum` du mot-clé `dce_iface` pour identifier le service pour l'opération.

Vous pouvez spécifier une valeur décimale unique comprise entre 0 et 65 535 pour une opération spécifique, une plage d'opérations séparées par un tiret ou une liste d'opérations et de plages séparées par des virgules, dans n'importe quel ordre.

N'importe lequel des exemples suivants spécifie des numéros d'opération de connexion réseau valides :

```
15
15-18
15, 18-20
15, 20-22, 17
15, 18-20, 22, 24-26
```

Le mot-clé dce_stub_data

Vous pouvez utiliser le mot-clé `dce_stub_data` avec le préprocesseur DCE/RPC pour spécifier que le moteur de règles doit commencer l'inspection au début des données tampons, quelles que soient les autres options de règle. Les options de règles de charge utile de paquet qui suivent le mot-clé `dce_stub_data` sont appliquées par rapport à la mémoire tampon de ces données.

Les données tampons DCE/RPC fournissent l'interface entre un appel de procédure client et le système d'exécution DCE/RPC, le mécanisme qui fournit les routines et les services essentiels à DCE/RPC. Les exploits DCE/RPC sont identifiés dans la partie données tampon du paquet DCE/RPC. Étant donné que les données

tampons sont associées à une opération ou à un appel de fonction spécifique, vous devez toujours faire précéder `dce_stub_data` de `dce_iface` et `dce_opnum` pour identifier le service et l'opération associés.

Le mot-clé `dce_stub_data` n'a aucun argument.

Mots-clés SIP

Quatre mots-clés SIP vous permettent de surveiller les exploits dans le trafic de session SIP.

Notez que le protocole SIP est vulnérable aux attaques par déni de service (DoS). Les règles qui traitent ces attaques peuvent bénéficier de la prévention des attaques basée sur le débit.

Le mot-clé sip_header

Vous pouvez utiliser le mot-clé `sip_header` pour commencer l'inspection au début de l'en-tête de demande ou de réponse SIP extrait et restreindre l'inspection aux champs d'en-tête.

Le mot-clé `sip_header` n'a pas d'argument.

L'exemple de fragment de règle suivant pointe vers l'en-tête SIP et correspond au champ d'en-tête CSeq :

```
alert udp any any -> any 5060 ( sip_header; content:"CSeq"; )
```

Sujets connexes

États des règles d'intrusion dynamique
Prévention des attaques basées sur le débit

Le mot-clé sip_body

Vous pouvez utiliser le mot-clé `sip_body` pour commencer l'inspection au début du corps du message de demande SIP ou de réponse extrait et restreindre l'inspection au corps du message.

Le mot-clé `sip_body` n'a pas d'argument.

L'exemple de fragment de règle suivant pointe vers le corps du message SIP et correspond à une adresse IP spécifique dans le champ c (connection information) des données SDP extraites :

```
alert udp any any -> any 5060 ( sip_body; content:"c=IN 192.168.12.14"; )
```

Notez que les règles ne se limitent pas à la recherche de contenu SDP. Le préprocesseur SIP extrait le corps entier du message et le met à la disposition du moteur de règles.

Le mot-clé sip_method

Un champ *méthode* dans chaque requête SIP identifie l'objectif de la demande. Vous pouvez utiliser le mot-clé `sip_method` pour tester les requêtes SIP de méthodes spécifiques. Séparez les valeurs de ports multiples par des virgules.

Vous pouvez spécifier l'une des méthodes SIP actuellement définies suivantes :

```
ack, benotify, bye, cancel, do, info, invite, join, message, notify, options, prack,
publish, quath, refer, register, service, sprack, subscribe, unsubscribe, update
```

Les méthodes sont insensibles à la casse. Vous pouvez séparer plusieurs méthodes par des virgules.

Étant donné que de nouvelles méthodes SIP pourraient être définies à l'avenir, vous pouvez également spécifier une méthode personnalisée, c'est-à-dire une méthode qui n'est pas une méthode SIP actuellement définie. Les valeurs de champ acceptées sont définies dans la RFC 2616, qui autorise tous les caractères à l'exception

des caractères de contrôle et des séparateurs comme `=`, `(` et `)`. Consultez la RFC 2616 pour obtenir la liste complète des séparateurs exclus. Lorsque le système rencontre une méthode personnalisée précisée dans le trafic, il inspecte l'en-tête du paquet, mais pas le message.

Le système prend en charge jusqu'à 32 méthodes, y compris les 21 méthodes actuellement définies et 11 autres méthodes. Le système ignore toutes les méthodes non définies que vous pourriez configurer. Notez que les 32 méthodes au total comprennent les méthodes spécifiées à l'aide de l'option **de méthodes de vérification du** préprocesseur SIP.

Vous ne pouvez spécifier qu'une seule méthode lorsque vous utilisez la négation. Par exemple :

```
!invite
```

Notez, cependant, que plusieurs mots-clés `sip_method` dans une règle sont liés à une opération **AND**. Par exemple, pour tester toutes les méthodes extraites à l'exception de `invite` et `cancel`, vous devez utiliser deux mots-clés de négation `sip_method` :

```
sip_method: !invite
sip_method: !cancel
```

Cisco vous recommande d'inclure au moins un mot-clé `content` dans les règles qui incluent le mot-clé `sip_method` pour vous assurer que le moteur de règles utilise le testeur de schéma rapide, ce qui accélère la vitesse de traitement et améliore les performances. Notez que le moteur de règles utilise la correspondance de modèle rapide lorsqu'une règle comprend au moins un mot-clé `content`, que vous ayez ou non activé l'argument **Use Fast Pattern Matcher** (utiliser la correspondance de modèle rapide) pour le mot-clé `content`.

Sujets connexes

Options du préprocesseur SIP
Les mots-clés content et protected_content, à la page 25
Arguments de la recherche de schéma rapide pour mot-clé de contenu, à la page 35

Le mot-clé sip_stat_code

Un code d'état à trois chiffres dans chaque réponse SIP indique le résultat de l'action demandée. Vous pouvez utiliser le mot-clé `sip_stat_code` pour tester les réponses SIP pour des codes d'état spécifiques.

Vous pouvez spécifier un nombre de type de réponse à un chiffre 1 à 9, un nombre à trois chiffres 100 à 999 ou une liste de n'importe quelle combinaison des deux éléments séparés par des virgules. Une liste correspond à un numéro de la liste qui correspond au code de la réponse SIP.

Le tableau suivant décrit les valeurs des codes d'état SIP que vous pouvez spécifier.

Tableau 53 : Valeurs `sip_stat_code`

Pour détecter...	Précisez...	Par exemple...	Détecte...
un code d'état précis	le code d'état à trois chiffres	189	189
tout code à trois chiffres commençant par un chiffre unique	le chiffre unique	1	1xx; c'est-à-dire 100,101, 102, etc.
une liste de valeurs	toute combinaison de codes spécifiques et de chiffres séparés par des virgules	222, 3	222 plus 300, 301, 302, etc.

Notez également que le moteur de règles n'utilise pas le match de modèle rapide pour rechercher la valeur précise à l'aide du mot-clé `sip_stat_code`, peu importe si votre règle comprend un mot-clé `content`.

Mots-clés GTP

Trois mots-clés de GSRP Tunneling Protocol (GTP) vous permettent d'inspecter le canal de commande GTP pour la version GTP, le type de message et les éléments d'information. Vous ne pouvez pas utiliser les mots-clés GTP en combinaison avec d'autres mots-clés de règles de prévention des intrusions tels que `content` ou `byte_jump`. Vous **devez** utiliser le mot-clé `gtp_version` dans chaque règle qui utilise le mot-clé `gtp_info` ou `gtp_type`.

Le mot-clé gtp_version

Vous pouvez utiliser le mot-clé `gtp_version` pour inspecter les messages de contrôle GTP à la recherche de la version 0, 1 ou 2.

Comme différentes versions de GTP définissent différents types de messages et éléments d'information, vous devez utiliser `gtp_version` lorsque vous utilisez le mot-clé `gtp_type` ou `gtp_info`. Vous pouvez spécifier la valeur 0, 1 ou 2.

Le mot-clé gtp_type

Chaque message GTP est identifié par un type de message, qui comprend une valeur numérique et une chaîne. Vous pouvez utiliser le mot-clé `gtp_type` pour inspecter le trafic à la recherche de types de messages GTP spécifiques. Comme différentes versions de GTP définissent différents types de messages et éléments d'information, vous devez également utiliser `gtp_version` lorsque vous utilisez le mot-clé `gtp_type` ou `gtp_info`.

Vous pouvez spécifier une valeur décimale définie pour un type de message, une chaîne définie ou une liste séparée par des virgules de l'un ou des deux, ou des deux, dans n'importe quelle combinaison, comme le montre l'exemple suivant :

```
10, 11, echo_request
```

Le système utilise une opération OU pour mettre en correspondance chaque valeur ou chaîne que vous répertoriez. L'ordre dans lequel vous répertoriez les valeurs et les chaînes n'a pas d'importance. Toute valeur ou chaîne unique de la liste correspond au mot-clé. Vous recevez une erreur si vous tentez d'enregistrer une règle qui comprend une chaîne non reconnue ou une valeur hors limites.

Notez dans le tableau que différentes versions de GTP utilisent parfois des valeurs différentes pour le même type de message. Par exemple, le type de message `sgsn_context_request` a une valeur de 50 dans GTPv0 et GTPv1, mais une valeur de 130 dans GTPv2.

Le mot-clé `gtp_type` correspond à différentes valeurs selon le numéro de version dans le paquet. Dans l'exemple ci-dessus, le mot-clé correspond à la valeur de type de message 50 dans un paquet GTPv0 ou GTPv1 et à la valeur à 130 dans un paquet GTPv2. Le mot-clé ne correspond pas à un paquet lorsque la valeur du type de message dans le paquet n'est pas une valeur connue pour la version spécifiée dans le paquet.

Si vous spécifiez un entier pour le type de message, le mot-clé correspond si le type de message dans le mot-clé correspond à la valeur du paquet GTP, quelle que soit la version spécifiée dans le paquet.

Le tableau suivant répertorie les valeurs définies et les chaînes reconnues par le système pour chaque type de message GTP.

Tableau 54 : Types de messages GTP

Valeur	Version 0	Version : 1	Version 2
1	echo_request	echo_request	echo_request
2	echo_response	echo_response	echo_response
3	version_not_supported	version_not_supported	version_not_supported
4	node_alive_request	node_alive_request	S.O.
5	node_alive_response	node_alive_response	S.O.
6	demande_redirection	demande_redirection	S. O.
7	réponse_redirection	réponse_redirection	S. O.
16	create_pdp_context_request	create_pdp_context_request	S. O.
17	create_pdp_context_response	create_pdp_context_response	S. O.
18	Update_pdp_context_request	Update_pdp_context_request	S. O.
19	update_pdp_context_response	update_pdp_context_response	S. O.
20	delete_pdp_context_request	delete_pdp_context_request	S. O.
21	delete_pdp_context_response	delete_pdp_context_response	S. O.
22	create_aa_pdp_context_request	init_pdp_context_activation_request	S.O.
23	create_aa_pdp_context_response	init_pdp_context_activation_response	S. O.
24	delete_aa_pdp_context_request	s.o.	s.o.
25	delete_aa_pdp_context_response	s.o.	s.o.
26	error_indication	error_indication	S. O.
27	pdu_notification_request	pdu_notification_request	S. O.
28	pdu_notification_response	pdu_notification_response	S. O.
29	pdu_notification_reject_request	pdu_notification_reject_request	S. O.
30	pdu_notification_reject_response	pdu_notification_reject_response	S. O.
31	S. O.	supported_ext_header_notification	S. O.
32	send_routing_info_request	send_routing_info_request	create_session_request
33	send_routing_info_response	send_routing_info_response	create_session_response
34	failure_report_request	failure_report_request	modify_bearer_request
35	failure_report_response	failure_report_response	modify_bearer_response

Valeur	Version 0	Version : 1	Version 2
36	note_ms_present_request	note_ms_présent_request	delete_session_request
37	note_ms_present_response	note_ms_present_response	delete_session_response
38	s.o.	s.o.	change_notification_request
39	s.o.	s.o.	change_notification_response
48	identification_request	identification_request	S. O.
49	identification_response	identification_response	S. O.
50	sgsn_context_request	sgsn_context_request	S. O.
51	sgsn_context_response	sgsn_context_response	S. O.
52	sgsn_context_ack	sgsn_context_ack	S. O.
53	S. O.	forward_relocation_request	S. O.
54	S. O.	forward_relocation_response	S. O.
55	S. O.	forward_relocation_complete	S.O.
56	S. O.	relocation_cancel_request	S. O.
57	S. O.	relocation_cancel_response	S. O.
58	S. O.	forward_srns_contex	S. O.
59	S. O.	forward_relocation_complete_ack	S. O.
60	S. O.	forward_srns_contex_ack	S. O.
64	s.o.	s.o.	commande_modifier_le_porteur
65	s.o.	s.o.	modify_bearer_failure_indication
66	s.o.	s.o.	delete_bearer_command
67	s.o.	s.o.	delete_bearer_failure_indication
68	s.o.	s.o.	bearer_resource_command
69	s.o.	s.o.	Bearer_resource_failure_indication
70	S. O.	ran_info_relay	descendant_failure_indication
71	s.o.	s.o.	trace_session_activation
72	s.o.	s.o.	trace_session_deactivation
73	s.o.	s.o.	stop_pages_indication
95	s.o.	s.o.	create_bearer_request

Valeur	Version 0	Version : 1	Version 2
96	S. O.	mbms_notification_request	create_bearer_response
97	S. O.	mbms_notification_response	update_bearer_request
98	S. O.	mbms_notification_reject_request	Update_bearer_response
99	S. O.	mbms_notification_reject_response	delete_bearer_request
100	S. O.	create_mbms_context_request	delete_bearer_response
101	S. O.	create_mbms_context_response	delete_pdn_request
102	S. O.	update_mbms_context_request	delete_pdn_response
103	S. O.	Update_mbms_context_response	S. O.
104	S. O.	delete_mbms_context_request	S. O.
105	S. O.	delete_mbms_context_response	S. O.
112	S. O.	mbms_register_request	S. O.
113	S. O.	mbms_register_response	S. O.
114	S. O.	mbms_deregister_request	S. O.
115	S. O.	mbms_deregister_response	S. O.
116	S. O.	mbms_session_start_request	S. O.
117	S. O.	mbms_session_start_response	S. O.
118	S. O.	mbms_session_stop_request	S. O.
119	S. O.	mbms_session_stop_response	S. O.
120	S. O.	mbms_session_update_request	S. O.
121	S. O.	mbms_session_update_response	S. O.
128	S. O.	ms_info_change_request	identification_request
129	S. O.	ms_info_change_response	identification_response
130	s.o.	s.o.	sgsn_context_request
131	s.o.	s.o.	sgsn_context_response
132	s.o.	s.o.	sgsn_context_ack
133	s.o.	s.o.	forward_relocation_request
134	s.o.	s.o.	forward_relocation_response
135	s.o.	s.o.	forward_relocation_complete

Valeur	Version 0	Version : 1	Version 2
136	s.o.	s.o.	forward_relocation_complete_ack
137	s.o.	s.o.	forward_access
138	s.o.	s.o.	forward_access_ack
139	s.o.	s.o.	relocation_cancel_request
140	s.o.	s.o.	relocation_cancel_response
141	s.o.	s.o.	configuration_transfer_tunnel
149	s.o.	s.o.	dissocier
150	s.o.	s.o.	detach_ack
151	s.o.	s.o.	cs_paging
152	s.o.	s.o.	ran_info_relay
153	s.o.	s.o.	alerte_mme
154	s.o.	s.o.	alert_mme_ack
155	s.o.	s.o.	ue_activity
156	s.o.	s.o.	ue_activity_ack
160	s.o.	s.o.	create_forward_tunnel_request
161	s.o.	s.o.	create_forward_tunnel_response
162	s.o.	s.o.	suspend
163	s.o.	s.o.	suspend_ack
164	s.o.	s.o.	reprendre
165	s.o.	s.o.	resume_ack
166	s.o.	s.o.	create_indirect_forward_tunnel_request
167	s.o.	s.o.	create_indirect_forward_tunnel_response
168	s.o.	s.o.	delete_indirect_forward_tunnel_request
169	s.o.	s.o.	delete_indirect_forward_tunnel_response
170	s.o.	s.o.	Release_access_bearer_request
171	s.o.	s.o.	Release_access_bearer_response
176	s.o.	s.o.	downlink_data
177	s.o.	s.o.	download_data_ack

Valeur	Version 0	Version : 1	Version 2
179	s.o.	s.o.	pgw_restart
180	s.o.	s.o.	pgw_restart_ack
200	s.o.	s.o.	Update_pdn_request
201	s.o.	s.o.	update_pdn_response
211	s.o.	s.o.	modify_access_bearer_request
212	s.o.	s.o.	modify_access_bearer_response
231	s.o.	s.o.	mbms_session_start_request
232	s.o.	s.o.	mbms_session_start_response
233	s.o.	s.o.	mbms_session_update_request
234	s.o.	s.o.	mbms_session_update_response
235.	s.o.	s.o.	mbms_session_stop_request
236	s.o.	s.o.	mbms_session_stop_response
240	data_record_transfer_request	data_record_transfer_request	S. O.
241	data_record_transfer_response	data_record_transfer_response	S. O.
254	S. O.	end_marker	S. O.
255	pdu	pdu	s.o.

Le mot-clé gtp_info

Un message GTP peut inclure plusieurs éléments d'information, chacun étant identifié à la fois par une valeur numérique et une chaîne définies. Vous pouvez utiliser le mot-clé `gtp_info` pour commencer l'inspection au début d'un élément d'information précisé et restreindre l'inspection à ce dernier. Comme différentes versions de GTP définissent différents types de messages et éléments d'information, vous devez également utiliser `gtp_version` lorsque vous utilisez ce mot-clé.

Vous pouvez spécifier la valeur décimale définie ou la chaîne définie pour un élément d'information. Vous pouvez spécifier une valeur ou une chaîne unique et utiliser plusieurs mots-clés `gtp_info` dans une règle pour inspecter plusieurs éléments d'information.

Lorsqu'un message comprend plusieurs éléments d'information du même type, tous sont examinés pour vérifier s'ils correspondent. Lorsque des éléments d'information apparaissent dans un ordre non valide, seule la dernière instance est inspectée.

À noter que différentes versions de GTP utilisent parfois des valeurs différentes pour le même élément d'information. Par exemple, l'élément d'information `cause` a la valeur 1 dans GTPv0 et GTPv1, mais la valeur 2 dans GTPv2.

Le mot-clé `gtp_info` correspond à différentes valeurs selon le numéro de version dans le paquet. Dans l'exemple ci-dessus, le mot-clé correspond à la valeur 1 de l'élément d'information dans un paquet GTPv0

ou GTPv1 et à la valeur 2 dans un paquet GTPv2. Le mot-clé ne correspond pas à un paquet lorsque la valeur de l'élément d'information dans le paquet n'est pas une valeur connue pour la version précisée dans le paquet.

Si vous spécifiez un entier pour l'élément d'information , le mot-clé correspond si le type de message dans le mot-clé correspond à la valeur dans le paquet GTP, quelle que soit la version spécifiée dans le paquet.

Le tableau suivant dresse la liste des valeurs et des chaînes reconnues par le système pour chaque élément d'information GTP.

Tableau 55 : Éléments d'information GTP

Valeur	Version 0	Version : 1	Version 2
1	cause	cause	imsi
2	imsi	imsi	cause
3	rai	rai	recovery
4	tlli	tlli	S.O.
5	p_tmsi	p_tmsi	S.O.
6	qos	s.o.	s.o.
8	recording_required	recording_required	S.O.
9	authentification	authentification	S.O.
11	map_cause	map_cause	S. O.
12	p_tmsi_sig	p_tmsi_sig	S. O.
13	ms_validated	ms_validated	S.O.
14	recovery	recovery	S.O.
15	selection_mode	selection_mode	S. O.
16	flow_label_data_1	teid_1	S. O.
17	flow_label_signalling	teid_control	S. O.
18	flow_label_data_2	teid_2	S. O.
19	ms_unreachable	teardown_ind	S. O.
20	S. O.	nsapi	S. O.
21	S. O.	ranap	S. O.
22	S. O.	rab_context	S.O.
23	S. O.	radio_priority_sms	S. O.
24	S. O.	radio_priority	S. O.

Valeur	Version 0	Version : 1	Version 2
25	S. O.	packet_flow_id	S. O.
26	S. O.	charging_char	S. O.
27	S. O.	trace_ref	S. O.
28	S. O.	trace_type	S. O.
29	S. O.	ms_unreachable	S. O.
71	s.o.	s.o.	apn
72	s.o.	s.o.	ambr
73	s.o.	s.o.	ebi
74	s.o.	s.o.	ip_addr
75	s.o.	s.o.	mei
76	s.o.	s.o.	msisdn
77	s.o.	s.o.	Indication
78	s.o.	s.o.	pco
79	s.o.	s.o.	paa
80	s.o.	s.o.	bearer_qos
80	s.o.	s.o.	flow_qos
82	s.o.	s.o.	rat_type
83	s.o.	s.o.	serving_network
84	s.o.	s.o.	bearer_tft
85	s.o.	s.o.	tad
86	s.o.	s.o.	uli
87	s.o.	s.o.	f_teid
88	s.o.	s.o.	tmsi
89	s.o.	s.o.	cn_id
90	s.o.	s.o.	s103pdf
91	s.o.	s.o.	s1udf
92	s.o.	s.o.	delay_value
93	s.o.	s.o.	bearer_context

Valeur	Version 0	Version : 1	Version 2
94	s.o.	s.o.	charging_id
95	s.o.	s.o.	charging_char
96	s.o.	s.o.	trace_info
97	s.o.	s.o.	bearer_flag
99	s.o.	s.o.	pdn_type
100	s.o.	s.o.	pti
101	s.o.	s.o.	drx_parameter
103	s.o.	s.o.	gsm_key_tri
104	s.o.	s.o.	umts_key_cipher_quin
105	s.o.	s.o.	gsm_key_cipher_quin
106	s.o.	s.o.	umts_key_quin
107	s.o.	s.o.	eps_quad
108	s.o.	s.o.	umts_key_quad_quin
109	s.o.	s.o.	pdn_connection
110	s.o.	s.o.	pdn_number
111	s.o.	s.o.	p_tmsi
112	s.o.	s.o.	p_tmsi_sig
113	s.o.	s.o.	hop_counter
114	s.o.	s.o.	ue_time_zone
115	s.o.	s.o.	trace_ref
116	s.o.	s.o.	complete_request_msg
117	s.o.	s.o.	guti
118	s.o.	s.o.	f_container
119	s.o.	s.o.	f_cause
120	s.o.	s.o.	plmn_id
121	s.o.	s.o.	target_id
123	s.o.	s.o.	packet_flow_id
124	s.o.	s.o.	rab_contex

Valeur	Version 0	Version : 1	Version 2
125	s.o.	s.o.	src_rnc_pdcp
126	s.o.	s.o.	udp_src_port
127	charge_id	charge_id	apn_restriction
128	end_user_address	end_user_address	selection_mode
129	mm_context	mm_context	src_id
130	pdp_context	pdp_context	S. O.
131	apn	apn	change_report_action
132	protocol_config	protocol_config	fq_csid
133	gsn	gsn	channel
134	msisdn	msisdn	emlpp_pri
135	S. O.	qos	node_type
136	S. O.	authentication_qu	fqdn
137	S. O.	tft	ti
138	S. O.	target_id	mbms_session_duration
139	S. O.	utran_trans	mbms_service_area
140	S. O.	rab_setup	mbms_session_id
141	S. O.	ext_header	mbms_flow_id
142	S. O.	trigger_id	mbms_ip_multicast
143	S. O.	omc_id	mbms_distribution_ack
144	S. O.	ran_trans	rfsp_index
145	S. O.	pdp_context_pri	uci
146	S. O.	addi_rab_setup	csg_info
147	S. O.	sgsn_number	csg_id
148	S. O.	common_flag	cmi
149	S. O.	apn_restriction	service_indicator
150	S. O.	radio_priority_lcs	detach_type
151	S. O.	rat_type	ldn
152	S. O.	user_loc_info	node_feature

Valeur	Version 0	Version : 1	Version 2
153	S. O.	ms_time_zone	mbms_time_to_transfer
154	S. O.	imei_sv	throttling
155	S. O.	camel	arp
156	S. O.	mbms_ue_context	epc_timer
157	S. O.	tmp_mobile_group_id	signalling_priority_indication
158	S. O.	rim_routing_addr	tmgi
159	S. O.	mbms_config	mm_srvcc
160	S. O.	mbms_service_area	flags_srvcc
161	S. O.	src_rnc_pdcp	nmbr
162	S. O.	addi_trace_info	S. O.
163	S. O.	hop_counter	S. O.
164	S. O.	plmn_id	S. O.
165	S. O.	mbms_session_id	S. O.
166	S. O.	mbms_2g3g_indicator	S. O.
167	S. O.	enhanced_nsapi	S. O.
168	S. O.	mbms_session_duration	S. O.
169	S. O.	addi_mbms_trace_info	S. O.
170	S. O.	mbms_session_repetition_num	S. O.
171	S. O.	mbms_time_to_data	S. O.
173	S. O.	bss	S. O.
174	S. O.	cell_id	S. O.
175	S. O.	pdu_num	S. O.
177	S. O.	mbms_bearer_capab	S. O.
178	S. O.	rim_routing_disc	S. O.
179	S. O.	list_pfc	S. O.
180	S. O.	ps_xid	S. O.
181	S. O.	ms_info_change_report	S. O.
182	S. O.	direct_tunnel_flags	S. O.

Valeur	Version 0	Version : 1	Version 2
183	S. O.	correlation_id	S. O.
184	S. O.	bearer_control_mode	S. O.
185	S. O.	mbms_flow_id	S. O.
186	S. O.	mbms_ip_multicast	S. O.
187	S. O.	mbms_distribution_ack	S. O.
188	S. O.	reliable_inter_rat_handover	S. O.
189	S. O.	rfsp_index	S. O.
190	S. O.	fqdn	S. O.
191	S. O.	evolved_allocation1	S. O.
192	S. O.	evolved_allocation2	S. O.
193	S. O.	extended_flags	S. O.
194	S. O.	uci	S. O.
195	S. O.	csg_info	S. O.
196	S. O.	csg_id	S. O.
197	S. O.	cmi	S. O.
198	S. O.	apn_ambr	S. O.
199	S. O.	ue_network	S. O.
200	S. O.	ue_ambr	S. O.
201	S. O.	apn_ambr_nsapi	S. O.
202	S. O.	ggsn_backoff_timer	S. O.
203	S. O.	signalling_priority_indication	S. O.
204	S. O.	signalling_priority_indication_nsapi	S. O.
205	S. O.	high_bitrate	S. O.
206	S. O.	max_mbr	S. O.
251	charging_gateway_addr	charging_gateway_addr	S. O.
255	private_extension	private_extension	private_extension

Mots-clés SCADA

Le moteur de règles utilise les règles Modbus, DNP3, CIP et S7Commplus pour accéder à certains champs de protocole.

Mots-clés Modbus

Vous pouvez utiliser les mots-clés Modbus seuls ou en combinaison avec d'autres mots-clés tels que `content` et `byte_jump`.

modbus_data

Vous pouvez utiliser le mot-clé `modbus_data` pour pointer vers le début du champ de données dans une requête ou réponse Modbus.

modbus_func

Vous pouvez utiliser le mot-clé `modbus_func` pour la mise en correspondance avec le champ Function Code dans une en-tête de demande ou de réponse de couche d'application Modbus. Vous pouvez spécifier une valeur décimale définie unique ou une chaîne définie unique pour un code de fonction Modbus.

Le tableau suivant répertorie les valeurs définies et les chaînes reconnues par le système pour les codes de fonction Modbus.

Tableau 56 : Codes de fonction Modbus

Valeur	Chaîne
1	read_coils
2	read_discrete_inputs
3	read_holding_registers
4	read_input_registers
5	write_single_coil
6	write_single_register
7	read_exception_status
8	diagnostics
11	get_comm_event_counter
12	get_comm_event_log
15	write_multiple_coils
16	write_multiple_registers
17	report_slave_id
20	read_file_record

Valeur	Chaîne
21	write_file_record
22	mask_write_register
23	read_write_multiple_registers
24	read_fifo_queue
43	encapsulated_interface_transport

modbus_unit

Vous pouvez utiliser le mot-clé `modbus_unit` pour faire correspondre une valeur décimale unique au champ Unit ID (ID d'unité) d'une demande ou d'un en-tête de réponse Modbus.

Mots-clés DNP3

Vous pouvez utiliser les mots-clés DNP3 seuls ou en combinaison avec d'autres mots-clés tels que `content` et `byte_jump`.

dnp3_data

Vous pouvez utiliser le mot-clé `dnp3_data` pour pointer vers le début des fragments de couche d'application DNP3 réassemblés.

Le préprocesseur DNP3 rassemble les trames de la couche de liaison en fragments de couche d'application. Le mot-clé `dnp3_data` pointe vers le début de chaque fragment de la couche d'application; d'autres options de règles peuvent être mises en correspondance avec les données réassemblées dans des fragments sans séparer les données et sans ajouter de sommes de contrôle tous les 16 octets.

dnp3_func

Vous pouvez utiliser le mot-clé `dnp3_func` pour la mise en correspondance avec le champ Function Code dans une en-tête de demande ou de réponse de couche d'application DNP3. Vous pouvez spécifier une valeur décimale définie unique ou une chaîne définie unique pour un code de fonction DNP3.

Le tableau suivant répertorie les valeurs définies et les chaînes reconnues par le système pour les codes de fonction DNP3.

Tableau 57 : Codes de fonction DNP3

Valeur	Chaîne
0	confirm
1	read
2	write
3	select
4	operate

Valeur	Chaîne
5	direct_operate
6	direct_operate_nr
7	immed_freeze
8	immed_freeze_nr
9	freeze_clear
10	freeze_clear_nr
11	freeze_at_time
12	freeze_at_time_nr
13	cold_restart
14	warm_restart
15	initialize_data
16	initialize_appl
17	start_appl
18	stop_appl
19	save_config
20	enable_unsolicited
21	disable_unsolicited
22	assign_class
23	delay_measure
24	record_current_time
25	open_file
26	close_file
27	delete_file
28	get_file_info
29	authenticate_file
30	abort_file
31	activate_config
32	authenticate_req

Valeur	Chaîne
33	authenticate_err
129	response
130	unsolicited_response
131	authenticate_resp

dnp3_ind

Vous pouvez utiliser le mot-clé `dnp3_ind` pour la mise en correspondance avec les indicateurs dans le champ Internal Indications (Indications internes) dans un en-tête de réponse de couche d'application DNP3.

Vous pouvez spécifier la chaîne pour un seul indicateur connu ou une liste d'indicateurs séparés par des virgules, comme le montre l'exemple suivant :

```
class_1_events, class_2_events
```

Lorsque vous spécifiez plusieurs indicateurs, le mot-clé correspond à n'importe quel indicateur de la liste. Pour détecter une combinaison d'indicateurs, utilisez le mot-clé `dnp3_ind` plusieurs fois dans une règle.

La liste suivante fournit la syntaxe de chaîne reconnue par le système pour les indicateurs d'indications internes DNP3 définis.

```
class_1_events
class_2_events
class_3_events
need_time
local_control
device_trouble
device_restart
no_func_code_support
object_unknown
parameter_error
event_buffer_overflow
already_executing
config_corrupt
reserved_2
reserved_1
```

dnp3_obj

Vous pouvez utiliser le mot-clé `dnp3_obj` pour une mise en correspondance avec les en-têtes d'objet DNP3 dans une demande ou une réponse.

Les données DNP3 sont composées d'une série d'objets DNP3 de différents types, comme une entrée analogique, une entrée binaire, etc. Chaque type est identifié par un *groupe*, par exemple un groupe d'entrées analogiques, un groupe d'entrées binaires, etc., chacun pouvant être identifié par une valeur décimale. Les objets de chaque groupe sont en outre identifiés par une *variation d'objet*, comme des entiers 16 bits, des entiers 32 bits, une virgule flottante courte, etc., chacun spécifiant le format de données de l'objet. Chaque type de variation d'objet peut également être identifié par une valeur décimale.

Vous définissez les en-têtes d'objets en précisant le numéro décimal du type de groupe d'en-têtes d'objet et le numéro décimal du type de variation d'objet. La combinaison des deux définit un type particulier d'objet DNP3.

Mots-clés CIP et ENIP

Vous pouvez utiliser les mots-clés suivants seuls ou en combinaison pour créer des règles de prévention des intrusions personnalisées qui identifient les attaques contre le trafic CIP et ENIP détecté par le préprocesseur CIP. Pour les mots-clés configurables, spécifiez un seul entier dans la plage autorisée. Consultez Le préprocesseur CIP pour obtenir de plus amples renseignements.

Tableau 58 :

Ce mot-clé...	correspond au/à la...	Plage
`cip_attribute`	champ Classe d'objet/attribut d'instance dans un message CIP. Précisez une valeur entière définie unique.	De 0 à 65535
`cip_class`	champ Object Class (classe d'objet) dans un message CIP. Précisez une valeur entière définie unique.	De 0 à 65535
`cip_conn_path_class`	la classe d'objet dans le chemin de connexion. Spécifiez une valeur entière unique.	De 0 à 65535
`cip_instance`	champ ID d'instance dans un message CIP. Spécifiez une valeur entière unique.	0 à 4284927295
`cip_req`	message de demande de service.	S. O.
`cip_rsp`	message de réponse de service.	S. O.
`service_cip`	champ Service dans un message de demande de service CIP. Spécifiez une valeur entière unique.	0 à 127
`cip_status`	champ Status (état) dans un message de réponse de service CIP. Spécifiez une valeur entière unique.	De 0 à 255
`enip_command`	code de commande dans l'en-tête EthNet/IP. Spécifiez une valeur entière unique.	De 0 à 65535
`enip_req`	message de demande EthNet/IP.	S. O.
`enip_rsp`	message de réponse EthNet/IP.	s.o.

Mots-clés S7Commplus

Vous pouvez utiliser les mots-clés S7Commplus seuls ou en combinaison pour créer des règles de prévention des intrusions personnalisées qui identifient les attaques de trafic détectées par le préprocesseur S7Commplus. Pour les mots-clés configurables, spécifiez une valeur unique connue ou un seul entier dans la plage autorisée. Consultez Le préprocesseur S7Commplus pour obtenir de plus amples renseignements.

Tenez compte des points suivants :

- Plusieurs mots-clés S7commplus de la même règle font l'objet d'une construction ET.

- L'utilisation de plusieurs mots-clés `s7commplus_func` ou `s7commplus_opcode` dans la même règle annulera la règle, et celle-ci ne correspondra jamais au trafic. Pour rechercher plusieurs valeurs avec ces mots clés, créez plusieurs règles.

s7commplus_content

Avant d'utiliser un mot-clé `content` ou `protected_content` dans une règle de prévention des intrusions S7Commplus, utilisez le mot-clé `s7commplus_content` pour positionner le curseur au début des données utiles du paquet. Consultez Les mots-clés content et protected_content, à la page 25 pour obtenir de plus amples renseignements.

s7commplus_func

Utilisez le mot-clé `s7commplus_func` pour faire la correspondance avec l'une des valeurs suivantes dans un en-tête S7Commplus :

- explore
- createobject
- deleteobject
- setvariable
- getlink
- setmultivar
- getmultivar
- beginsequence
- endsequence
- invoke
- getvarsubstr
- 0x0 through 0xFFF

  Notez que les expressions numériques permettent des valeurs supplémentaires.

s7commplus_opcode

Utilisez le mot-clé `s7commplus_opcode` pour faire la correspondance avec l'une des valeurs suivantes dans un en-tête S7Commplus :

- demande
- response
- notification
- response2
- 0x0 through 0xFF

  Notez que les expressions numériques permettent des valeurs supplémentaires.

Caractéristiques des paquets

Vous pouvez écrire des règles qui génèrent des événements uniquement sur des paquets ayant des caractéristiques de paquets spécifiques.

dsize

Le mot-clé `dsize` teste la taille de la charge utile du paquet. Grâce à lui, vous pouvez utiliser les opérateurs supérieur à et inférieur à (< et >) pour spécifier une plage de valeurs. Vous pouvez utiliser la syntaxe suivante pour spécifier des plages :

```
>number_of_bytes
<number_of_bytes
number_of_bytes<>number_of_bytes
```

Par exemple, pour indiquer une taille de paquet supérieure à 400 octets, utilisez `>400` comme valeur `dtype`. Pour indiquer une taille de paquet de moins de 500 octets, utilisez `<500`. Pour spécifier que la règle se déclenche sur n'importe quel paquet dont la taille est comprise entre 400 et 500 octets, utilisez `400<>500|`

Mise en garde Le mot-clé `dsize` teste les paquets avant qu'ils ne soient décodés par des préprocesseurs.

isdataat

Le mot-clé `isdataat` demande au moteur de règles de vérifier que les données se trouvent à un emplacement spécifique dans la charge utile.

Le tableau suivant répertorie les arguments que vous pouvez utiliser avec le mot-clé `isdataat`.

Tableau 59 : Arguments isdataat

Argument	Type	Description
Décalage	Obligatoire	L'emplacement spécifique dans la charge utile. Par exemple, pour tester que les données apparaissent à l'octet 50 dans la charge utile du paquet, vous devez spécifier `50` comme valeur de décalage. Un modificateur `!` annulera les résultats du test `isdataat` ; il alerte si une certaine quantité de données n'est pas présente dans la charge utile. Vous pouvez également utiliser une variable `byte_extract` existante ou un résultat `byte_math` pour spécifier la valeur de cet arguments.
Relatif	Facultatif	Relie l'emplacement à la dernière correspondance de contenu réussie. Si vous spécifiez un emplacement relatif, notez que le compteur commence à l'octet 0. Calculez donc l'emplacement en soustrayant 1 du nombre d'octets dont vous souhaitez avancer à partir de la dernière correspondance de contenu réussie. Par exemple, pour spécifier que les données doivent apparaître au neuvième octet après la dernière correspondance de contenu réussie, vous devez spécifier un décalage relatif de `8`.
Données brutes	Facultatif	Spécifie que les données se trouvent dans la charge utile du paquet d'origine avant le décodage ou la normalisation de la couche d'application par un préprocesseur du système Firepower. Vous pouvez utiliser cet arguments avec **relative** si la correspondance de contenu précédente se trouve dans les données brutes du paquet.

Par exemple, dans une règle recherchant le contenu `toto`, si la valeur de `isdataat` est spécifiée comme suit :

- Offset = !10
- Relative = enabled

Le système alerte si le moteur de règles ne détecte pas 10 octets après `toto` avant la fin de la charge utile.

sameip

Le mot-clé `sameip` teste que les adresses IP de source et de destination d'un paquet sont identiques. Il ne nécessite pas d'argument.

fragoffset

Le mot-clé `fragoffset` teste le décalage d'un paquet fragmenté. Cela est utile, car certaines exploitations (telles que les attaques par déni de service WindowsNUK) utilisent des fragments de paquets générés manuellement qui ont des décalages spécifiques.

Par exemple, pour tester si le décalage d'un paquet fragmenté est de 31337 octets, spécifiez `31337` comme valeur de `fragoffset`.

Vous pouvez utiliser les opérateurs suivants lors de la spécification des arguments du mot-clé `fragoffset`.

Tableau 60 : Opérateurs d'arguments de mot-clé fragoffset

Opérateur	Description
!	pas
>	supérieur à
<	inférieur à

Notez que vous ne pouvez pas utiliser l'opérateur not (`!`) en combinaison avec < ou >.

cvs

Le mot-clé `cvs` teste le trafic CVS (Concurrent Versions System) à la recherche d'entrées CVS mal formées. Un attaquant peut utiliser une entrée malformée pour provoquer un débordement de tas et exécuter du code malveillant sur le serveur CVS. Ce mot-clé peut être utilisé pour identifier des attaques contre deux vulnérabilités CVS connues : CVE-2004-0396 (CVS 1.11.x jusqu'à 1.11.15 et 1.12.x jusqu'à 1.12.7) et CVS-2004-0414 (CVS 1.12. x à 1.12.8 et 1.11.x à 1.11.16). Le mot-clé `cvs` vérifie si une entrée est bien formée et génère des alertes lorsqu'une entrée mal formée est détectée.

Votre règle doit inclure les ports sur lesquels CVS est exécuté. En outre, tous les ports où le trafic peut se produire doivent être ajoutés à la liste des ports pour le réassemblage des flux dans vos politiques TCP afin que l'état puisse être maintenu pour les sessions CVS. Les ports TCP 2401 (`pserver`) et 514 (`rsh`) sont inclus dans la liste des ports clients où le réassemblage des flux a lieu. Cependant, notez que si votre serveur fonctionne en tant que serveur `xinetd` (c.-à-d., pserver), il peut fonctionner sur n'importe quel port TCP. Ajoutez tous les ports non standard à la liste des **ports client** de réassemblage de flux.

Sujets connexes

Le mot-clé byte_extract, à la page 43
Options de prétraitement du flux TCP

Mots-clés de la réponse active

Les mots-clés **resp** et **react** offrent deux approches pour lancer des réponses actives. Une règle de prévention des intrusions qui contient l'un ou l'autre de ces mots-clés déclenche une seule réponse active lorsqu'un paquet déclenche la règle. Les mots-clés de réponse active déclenchent des réponses actives pour fermer les connexions TCP en réponse aux règles TCP déclenchées ou les sessions UDP en réponse aux règles UDP déclenchées. Consultez Réponses actives dans les règles de suppression de prévention des intrusions. Les réponses actives ne sont pas destinées à remplacer un pare-feu pour un certain nombre de raisons, notamment le fait qu'un agresseur peut avoir choisi d'ignorer ou de contourner les réponses actives.

Les réponses actives sont prises en charge dans les déploiements en ligne, y compris les déploiements routés ou transparents. Par exemple, en réponse au mot-clé `react` dans un déploiement en ligne, le système peut insérer un paquet de réinitialisation TCP (RST) directement dans le trafic à chaque extrémité de la connexion, ce qui devrait normalement la fermer. Les réponses actives ne sont pas prises en charge ou ne conviennent pas aux déploiements passifs.

Comme les réponses actives peuvent être routées en retour, le système ne permet pas aux réinitialisations TCP de lancer des réinitialisations TCP; cela empêche une séquence sans fin de réponses actives. Le système ne permet pas non plus aux paquets ICMP inaccessibles de lancer des paquets ICMP inaccessibles, conformément à la pratique courante.

Vous pouvez configurer le préprocesseur de flux TCP pour détecter le trafic supplémentaire sur une connexion TCP après qu'une règle de prévention des intrusions a déclenché une réponse active. Lorsque le préprocesseur détecte du trafic supplémentaire, il envoie des réponses actives supplémentaires jusqu'à un maximum spécifié aux deux extrémités de la connexion ou de la session. Voir **Nombre maximal de réponses actives** et Nombre **minimal de secondes de réponse** dans Options avancées de préprocesseur transport/réseau.

Sujets connexes

Réponses actives dans les règles de suppression de prévention des intrusions

Le mot-clé resp

Vous pouvez utiliser le mot-clé `resp` pour répondre activement aux connexions TCP ou aux sessions UDP, selon que vous spécifiez le protocole TCP ou UDP dans l'en-tête de règle.

Les arguments de mots clés vous permettent de préciser la direction des paquets et d'utiliser les paquets de réinitialisation TCP (RST) ou ICMP inaccessible comme réponses actives.

Vous pouvez utiliser n'importe lequel des arguments de réinitialisation TCP ou ICMP inaccessible pour clore les connexions TCP. Vous devez utiliser uniquement des arguments ICMP inaccessible pour clore les sessions UDP.

différents arguments de réinitialisation TCP vous permettent également de cibler les réponses actives à la source du paquet, à la destination ou aux deux. Tous les arguments ICMP inaccessible ciblent la source du paquet et vous permettent de spécifier s'il faut utiliser un réseau ICMP, un hôte ou un paquet de port inaccessible, ou les trois.

Le tableau suivant répertorie les arguments que vous pouvez utiliser avec le mot-clé `resp` pour spécifier exactement ce que vous voulez que le système Firepower fasse lorsque la règle se déclenche.

Tableau 61 : Arguments resp

Argument	Description
reset_source	Dirige un paquet de réinitialisation TCP vers le point terminal qui a envoyé le paquet qui a déclenché la règle. Vous pouvez également définir `rst_snd`qui est pris en charge à des fins de compatibilité ascendante.

Argument	Description
reset_dest	Dirige un paquet de réinitialisation TCP vers le point terminal de destination du paquet qui a déclenché la règle. Vous pouvez également spécifier `rst_rcv`, qui est pris en charge pour la compatibilité ascendante.
reset_both	Dirige un paquet de réinitialisation TCP vers les points terminaux expéditeur et destinataire. Vous pouvez également définir `rst_al`, qui est pris en charge pour des raisons de compatibilité ascendante.
icmp_net	Dirige un message ICMP network unreachable (« Réseau ICMP inaccessible ») vers l'expéditeur.
hôte_icmp	Envoie un message « Hôte ICMP inaccessible » vers l'expéditeur.
icmp_port	Envoie un message « Port ICMP inaccessible » vers l'expéditeur. Cet arguments est utilisé pour mettre fin au trafic UDP.
icmp_all	Dirige les messages ICMP suivants vers l'expéditeur : • network unreachable (réseau inaccessible) • host unreachable (hôte inaccessible) • port unreachable (port inaccessible)

Par exemple, pour configurer une règle afin de réinitialiser les deux côtés d'une connexion lorsqu'une règle est déclenchée, utilisez `reset_both` comme valeur pour le mot-clé `resp`.

Vous pouvez utiliser une liste séparée par des virgules pour spécifier plusieurs arguments comme suit :

```
argument,argument,argument
```

Le mot-clé react (réaction)

Vous pouvez utiliser le mot-clé `react` pour envoyer une page HTML par défaut au client de connexion TCP lorsqu'un paquet déclenche la règle; après l'envoi de la page HTML, le système utilise les paquets de réinitialisation TCP pour initier des réponses actives aux deux extrémités de la connexion. Le mot-clé `react` ne déclenche pas de réponses actives pour le trafic UDP.

Vous pouvez également spécifier l'argument suivant :

```
msg
```

Lorsqu'un paquet déclenche une règle `react` qui utilise l'argument `msg`, la page HTML inclut le message d'événement de règle.

Si vous ne spécifiez pas d'argument `msg`, la page HTML comprend le message suivant :

```
You are attempting to access a forbidden site.
Consult your system administrator for details.
```

Remarque Étant donné que les réponses actives peuvent être routées en retour, vérifiez que la page de réponse HTML ne déclenche pas de règle `react`; cela pourrait entraîner une séquence interminable de réponses actives. Cisco vous recommande de tester les règles `react` de manière approfondie avant de les activer dans un environnement de production.

Sujets connexes

Anatomie des règles, à la page 3

Le mot-clé detection_filter

Vous pouvez utiliser le mot-clé `detection_filter` pour empêcher une règle de générer des événements, sauf si un nombre spécifié de paquets déclenche la règle dans un délai spécifié. Cela peut empêcher la règle de générer prématurément des événements. Par exemple, deux ou trois tentatives de connexion infructueuses en quelques secondes peuvent être un comportement attendu, mais un grand nombre de tentatives effectuées dans le même temps peut indiquer une attaque par force brute.

Le mot-clé `detection_filter` nécessite des arguments qui définissent si le système suit l'adresse IP source ou de destination, le nombre de fois que les critères de détection doivent être remplis avant de déclencher un événement et combien de temps le décompte doit-il continuer.

Utilisez la syntaxe suivante pour retarder le déclenchement des événements :

```
track by_src/by_dst, count count, seconds number_of_seconds
```

L'argument `track` spécifie s'il faut utiliser l'adresse IP de source ou de destination du paquet lors du comptage du nombre de paquets qui répondent aux critères de détection de la règle. Sélectionnez une des valeurs d'arguments décrites dans le tableau suivant pour préciser comment le système suit les instances d'événement.

Tableau 62 : Arguments du suivi detection_filter

Argument	Description
by_src	Nombre de critères de détection par adresse IP source.
by_dst	Nombre de critères de détection par adresse IP de destination.

L'argument `count` précise le nombre de paquets qui doivent déclencher la règle pour l'adresse IP précisée dans le délai précisé avant que la règle génère un événement.

L'argument des `seconds` précise le nombre de secondes pendant lesquelles le nombre de paquets doit déclencher la règle avant que la règle ne génère un événement.

Prenons le cas d'une règle qui recherche dans les paquets le contenu `foo` et utilise le mot-clé `detection_filter` avec les arguments suivants :

```
track by_src, count 10, seconds 20
```

Dans l'exemple, la règle ne générera pas d'événement tant qu'elle n'aura pas détecté `foo` dans 10 paquets en 20 secondes à partir d'une adresse IP source donnée. Si le système détecte seulement 7 paquets contenant `foo` dans les 20 premières secondes, aucun événement n'est généré. Toutefois, si `foo` se produit 40 fois dans les 20 premières secondes, la règle génère 30 événements et le décompte recommence lorsque 20 secondes se sont écoulées.

Comparaison des mots clés de seuil et detection_filter

Le mot-clé `detection_filter` remplace le mot-clé obsolète `threshold` (seuil). Le mot-clé de `threshold` est toujours pris en charge pour la compatibilité en amont et fonctionne de la même façon que les seuils que vous définissez dans une politique de prévention des intrusions.

Le mot-clé `detection_filter` est une fonctionnalité de détection qui est appliquée avant qu'un paquet ne déclenche une règle. La règle ne génère pas d'événement pour déclencher les paquets détectés avant le nombre de paquets spécifié et, dans un déploiement en ligne, ne supprime pas ces paquets si la règle est définie pour abandonner des paquets. Inversement, la règle génère des événements pour les paquets qui déclenchent la règle et se produisent après le nombre de paquets spécifié et, dans un déploiement en ligne, supprime ces paquets si la règle est définie pour abandonner des paquets.

Le seuil est une fonctionnalité de notification d'événement qui n'entraîne pas de détection. Elle est appliquée après qu'un paquet a déclenché un événement. Dans un déploiement en ligne, une règle définie pour supprimer les paquets supprime tous les paquets qui déclenchent la règle, quel que soit le seuil de règle.

Notez que vous pouvez utiliser le mot-clé `detection_filter` dans n'importe quelle combinaison avec les fonctions de fixation de seuil des incidents d'intrusion, de suppression des incidents d'intrusion et de prévention des attaques basée sur le débit d'une politique de prévention des intrusions. La validation de la politique échoue si vous activez une règle locale importée qui utilise le mot-clé `threshold` (seuil) obsolète en combinaison avec la fonction de seuillage des incidents d'intrusion dans une politique de prévention des intrusions.

Sujets connexes

Seuils de incidents d'intrusion
Configuration de la suppression des politiques de prévention des intrusions
Définition d'un état de règle dynamique à partir de la page Rules (Règles)

Le mot-clé tag

Utilisez le mot-clé `tag` pour demander au système de consigner le trafic supplémentaire pour l'hôte ou la session. Utilisez la syntaxe suivante lorsque vous spécifiez le type et le volume de trafic que vous souhaitez capter à l'aide du mot-clé `tag` :

```
tagging_type, count, metric, optional_direction
```

Les trois tableaux suivants décrivent les autres arguments disponibles.

Vous avez le choix entre deux types de balisage. Le tableau suivant décrit les deux types de balisage. Notez que le type d'argument de balise de session permet au système de consigner les paquets de la même session comme s'ils provenaient de sessions différentes si vous configurez uniquement les options d'en-tête de règle dans la règle de prévention des intrusions. Pour regrouper des paquets d'une même session, configurez une ou plusieurs options de règle (comme un mot-clé `flag` ou un mot-clé de `content`) dans la même règle de prévention des intrusions.

Tableau 63 : Arguments de balise

Argument	Description
séance de formation	Enregistre les paquets dans la session qui a déclenché la règle.
hôte	Consigne les paquets de l'hôte qui a envoyé le paquet qui a déclenché la règle. Vous pouvez ajouter un modificateur directionnel pour journaliser uniquement le trafic provenant de l'hôte (`src`) ou se rendant à l'hôte (`dst`).

Pour indiquer le volume de trafic que vous souhaitez consigner, utilisez l'argument suivant :

Tableau 64 : Nombre d'arguments

Argument	Description
Nombre	Le nombre de paquets ou de secondes que vous souhaitez journaliser après le déclenchement de la règle. Cette unité de mesure est spécifiée avec l'argument métrique, qui suit l'argument nombre.

Sélectionnez la mesure que vous souhaitez utiliser pour la journalisation en fonction de la durée ou du volume du trafic parmi celles décrites dans le tableau suivant.

Mise en garde Les réseaux à bande passante élevée peuvent voir des milliers de paquets par seconde, et le marquage d'un grand nombre de paquets peut sérieusement affecter les performances, alors assurez-vous d'ajuster ce paramètre pour votre environnement réseau.

Tableau 65 : Arguments des mesures de journalisation

Argument	Description
paquets	Consigne le nombre de paquets spécifié par le nombre après les déclenchements de la règle.
secondes	Consigne le trafic pendant le nombre de secondes spécifiée par le nombre après le déclenchement de la règle.

Par exemple, lorsqu'une règle avec la valeur de mot-clé `tag` suivante se déclenche :

```
host, 30, seconds, dst
```

tous les paquets transmis du client à l'hôte pendant les 30 prochaines secondes sont journalisés.

Le mot-clé flowbits

Utilisez le mot-clé `flowbits` pour affecter des noms d'état aux sessions. En analysant les paquets suivants dans une session en fonction de l'état nommé précédemment, le système peut détecter les exploits qui couvrent plusieurs paquets au cours d'une seule session et envoyer des alertes.

Le nom d'état `flowbits` est une étiquette définie par l'utilisateur attribuée aux paquets dans une partie spécifique d'une session. Vous pouvez étiqueter les paquets avec des noms d'état en fonction de leur contenu pour aider à distinguer les paquets malveillants de ceux pour lesquels vous ne souhaitez pas envoyer d'alerte. Vous pouvez définir jusqu'à 1 024 noms d'état par périphérique géré. Par exemple, si vous souhaitez recevoir une alerte sur les paquets malveillants qui ne se produisent qu'après une connexion réussie, vous pouvez utiliser le mot-clé `flowbits` pour filtrer les paquets qui constituent une tentative de connexion initiale afin de pouvoir vous concentrer uniquement sur les paquets malveillants. Vous pouvez le faire en créant d'abord une règle qui étiquettes tous les paquets de la session qui ont une connexion établie avec un état `Log_in`, puis en créant une deuxième règle dans laquelle `flowbits` vérifie les paquets avec l'état que vous avez défini dans la première règle et agit uniquement sur ceux-ci .

Un *nom de groupe* facultatif vous permet d'inclure un nom d'état dans un groupe d'états. Un nom d'état peut appartenir à plusieurs groupes. Les états non associés à un groupe ne s'excluent pas mutuellement. Par conséquent, une règle qui déclenche et définit un état qui n'est pas associé à un groupe n'affecte pas les autres états actuellement définis.

Options du mot-clé flowbits

Le tableau suivant décrit les différentes combinaisons d'opérateurs, d'états et de groupes disponibles pour le mot-clé `flowbits`. Notez que les noms d'état peuvent contenir des caractères alphanumériques, des points (.), des traits de soulignement (_) et des tirets (-).

Tableau 66 : Options de flowbits

Opérateur	Option d'état	Groupe	Description
`set`	`state_name`	Facultatif	Définit l'état spécifié pour un paquet. Définit l'état dans le groupe spécifié si un groupe est défini.
`set`	`state_name&state_name`	Facultatif	Définit les états spécifiés pour un paquet. Définit les états dans le groupe spécifié si un groupe est défini.
`setx`	`state_name`	obligatoire	Définit l'état précisé dans le groupe précisé pour un paquet et annule l'activation de tous les autres états du groupe.
`setx`	`state_name&state_name`	obligatoire	Définit les états précisés dans le groupe précisé pour un paquet et annule l'activation de tous les autres états du groupe.
`unset`	`state_name`	aucun groupe	Annule l'état spécifié pour un paquet.
`unset`	`state_name&state_name`	aucun groupe	Annule les états spécifiés pour un paquet.
`unset`	`all`	obligatoire	Annule tous les états dans le groupe spécifié.
`toggle`	`state_name`	aucun groupe	Annule l'état spécifié s'il est défini et définit l'état spécifié s'il l'est.
`toggle`	`state_name&state_name`	aucun groupe	Annule les états spécifiés s'ils sont définis et définit les états spécifiés s'ils le sont.
`toggle`	`all`	obligatoire	Désactive tous les états définis dans le groupe spécifié et désactive tous les états dans le groupe spécifié.
`isset`	`state_name`	aucun groupe	Détermine si l'état spécifié est défini dans le paquet.
`isset`	`state_name&state_name`	aucun groupe	Détermine si les états spécifiés sont définis dans le paquet.
`isset`	`state_name|state_name`	aucun groupe	Détermine si l'un des états spécifiés est défini dans le paquet.

Opérateur	Option d'état	Groupe	Description
`isset`	`any`	obligatoire	Détermine si un état est défini dans le groupe spécifié.
`isset`	`all`	obligatoire	Détermine si tous les états sont définis dans le groupe spécifié.
`isnotset`	`state_name`	aucun groupe	Détermine si l'état spécifié n'est pas défini dans le paquet.
`isnotset`	`state_name&state_name`	aucun groupe	Détermine si les états spécifiés ne sont pas définis dans le paquet.
`isnotset`	`state_name|state_name`	aucun groupe	Détermine si l'un des états spécifiés n'est pas défini dans le paquet.
`isnotset`	`any`	obligatoire	Détermine si un état n'est pas défini dans le paquet.
`isnotset`	`all`	obligatoire	Détermine si tous les états ne sont pas définis dans le paquet.
`reset`	(sans état)	Facultatif	Annule tous les états pour tous les paquets. Annule tous les états dans un groupe si un groupe est spécifié.
`noalert`	(sans état)	aucun groupe	Utilisez-le conjointement avec un autre opérateur pour supprimer la génération d'événements.

Lignes directrices pour l'utilisation du mot-clé flowbits

Tenez compte des éléments suivants lorsque vous utilisez le mot-clé `flowbits` :

- Lorsque vous utilisez l'opérateur `setx`, l'état spécifié ne peut appartenir qu'au groupe spécifié et à aucun autre groupe.
- Vous pouvez définir l'opérateur `setx` plusieurs fois, en spécifiant différents états et le même groupe à chaque instance.
- Lorsque vous utilisez l'opérateur `setx` et spécifiez un groupe, vous ne pouvez pas utiliser les opérateurs `set`, `toggle` ou `unset` sur ce groupe spécifié.
- Les opérateurs `isset` et `isnotset` évaluent pour l'état spécifié, peu importe si l'état se trouve dans un groupe.
- Pendant l'enregistrement de la politique de prévention des intrusions, la politique de prévention des intrusions s'applique de nouveau et la politique de contrôle d'accès s'applique (peu importe si la politique de contrôle d'accès fait référence à une ou à plusieurs politiques de prévention des intrusions), si vous activez une règle qui contient l'opérateur `isset` ou `isnotset` **sans** groupe précisé, et vous n'activez pas au moins une règle qui affecte l'affectation `de bits de flux` (`set`, `setx`, `unset`, `toggle`) pour le nom d'état et le protocole correspondants, toutes les règles qui affectent l'affectation de `flowbits` pour le nom d'état correspondant sont activées.
- Pendant les enregistrements de la politique de prévention des intrusions, la politique de prévention des intrusions s'applique de nouveau et que la politique de contrôle d'accès s'applique (peu importe si la politique de contrôle d'accès fait référence à une politique de prévention des intrusions ou à plusieurs

politiques de prévention des intrusions), si vous activez une règle qui contient l'opérateur `isset` ou `isnotset` **à** un groupe précisé, tous les règles qui affectent l'affectation de `flowbits`, (`set`, `setx`, `unset`, `toggle`) et définissent un nom de groupe correspondant sont également activées.

Exemples de mots-clés flowbits

Cette section fournit trois exemples qui utilisent le mot-clé `flowbits`.

Exemple de mot-clé flowbits : configuration A à l'aide de state_name

Ceci est un exemple de configuration flowbits utilisant `state_name`.

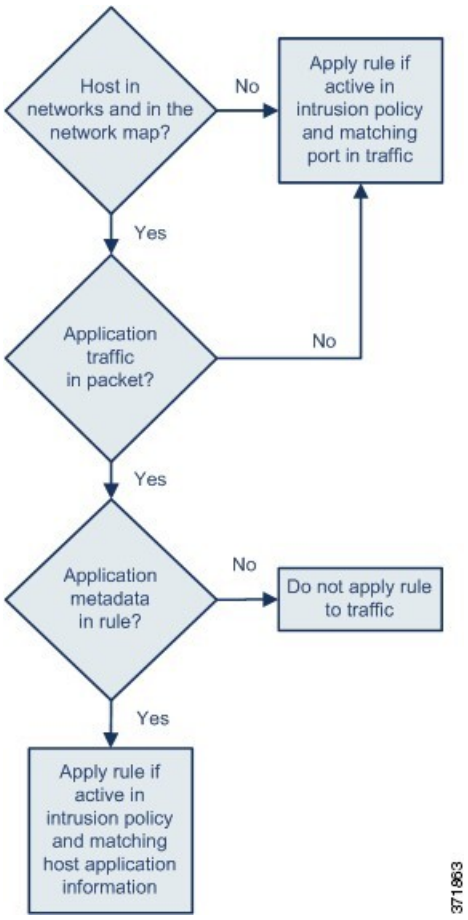
Prenez en compte la vulnérabilité IMAP décrite dans l'ID CVE 2000-0284. Cette vulnérabilité existe dans une implémentation d'IMAP, particulièrement dans les commandes LIST, LSUB, RENAME, FIND et COPY. Cependant, pour profiter de cette vulnérabilité, l'agresseur doit être connecté au serveur IMAP. Étant donné que la confirmation de connexion du serveur IMAP et l'exploitation qui suit se trouvent nécessairement dans des paquets différents, il est difficile de construire des règles non basées sur le flux qui détectent cette exploitation. À l'aide du mot-clé `flowbits`, vous pouvez cflowbitcréer une série de règles qui déterminent si l'utilisateur est connecté au serveur IMAP et, si c'est le cas, génèrent un événement si l'une des attaques est détectée. Si l'utilisateur n'est pas connecté, l'attaque ne peut pas exploiter la vulnérabilité et aucun événement n'est généré.

Les deux fragments de règle qui suivent illustrent cet exemple. Le premier fragment de règle recherche une confirmation de connexion IMAP du serveur IMAP :

```
alert tcp any 143 -> any any (msg:"IMAP login"; content:"OK

LOGIN"; flowbits:set,logged_in; flowbits:noalert;)
```

Le diagramme suivant illustre l'effet du mot-clé `flowbits` dans le fragment de règle précédent :



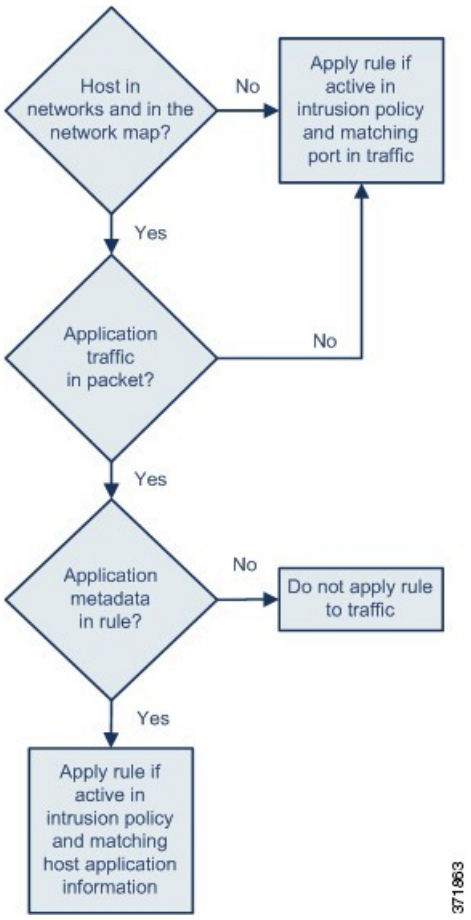
Notez que `flowbits:set` définit l'état `Logged_in`, tandis que `flowbits:noalert` supprime l'alerte, car vous verrez probablement de nombreuses sessions de connexion inoffensives sur un serveur IMAP.

Le fragment de règle suivant recherche une chaîne LISTE, mais ne génère pas d'événement à moins que l'état logged_in ait été défini à la suite d'un paquet précédent de la session :

```
alert tcp any any -> any 143 (msg:"IMAP LIST";

content:"LIST"; flowbits:isset,logged_in;)
```

Le diagramme suivant illustre l'effet du mot-clé `flowbits` dans le fragment de règle précédent :



Dans ce cas, si un paquet précédent a entraîné le déclenchement d'une règle contenant le premier fragment, une règle contenant le deuxième fragment se déclenche et génère un événement.

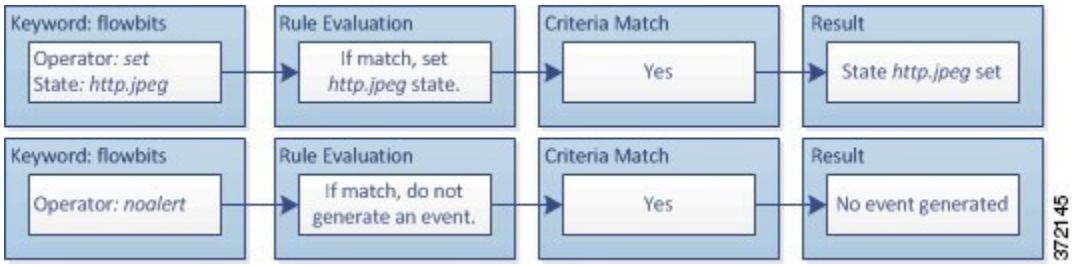
Exemple de mot-clé flowbits : configuration A entraînant des événements faux positifs

L'inclusion de noms d'état différents qui sont définis dans différentes règles d'un groupe peut éviter les événements faux positifs qui pourraient se produire lorsque le contenu d'un paquet suivant correspond à une règle dont l'état n'est plus valide. L'exemple suivant illustre comment vous pouvez obtenir de faux positifs lorsque vous n'incluez pas plusieurs noms d'état dans un groupe.

Voici le cas où les trois fragments de règle suivants se déclenchent dans l'ordre indiqué au cours d'une seule session :

```
(msg:"JPEG transfer";
content:"image/";pcre:"/^Content-?Type\x3a(\s*|\s*\r?\n\s+)image\x2fp?jpe?g/smi";
?flowbits:set,http.jpeg; flowbits:noalert;)
```

Le diagramme suivant illustre l'effet du mot-clé `flowbits` dans le fragment de règle précédent :

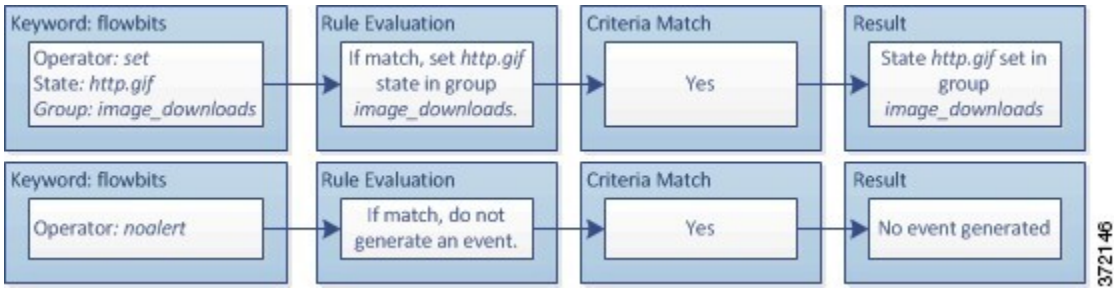


Les mots-clés `content` et `pcre` dans le premier fragment de règle correspondent à un téléchargement de fichier JPG, `flowbits:set,http.jpeg` définit l'état `flowbits http.jpeg` et `flowbits:noalert` empêche la règle de générer des événements. Aucun événement n'est généré, car l'objectif de la règle est de détecter le téléchargement de fichier et de définir l'état `flowbits` de sorte qu'une ou plusieurs règles associées peuvent tester le nom d'état associé au contenu malveillant et générer des événements lorsqu'un contenu malveillant est détecté.

Le fragment de règle suivant détecte un téléchargement de fichier GIF à la suite du téléchargement de fichier jpeg ci-dessus :

```
(msg:"GIF transfer"; content:"image/";
pcre:"/^Content-?Type\x3a(\s*|\s*\r?\n\s+)image\x2fgif/smi";
?flowbits:set,http.jpg,image_downloads; flowbits:noalert;)
```

Le diagramme suivant illustre l'effet du mot-clé `flowbits` dans le fragment de règle précédent :

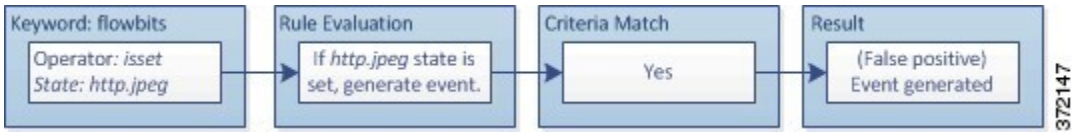


Les mots-clés `content` et `pcre` de la deuxième règle correspondent au téléchargement du fichier GIF, `flowbits:set,http.jpg` définit l'état du flowbit `http.jpg` et `flowbits:noalert` empêche la règle de générer un événement. Notez que l'état `http.jpeg` défini par le premier fragment de règle est toujours défini même s'il n'est plus nécessaire; en effet, le téléchargement du fichier jpeg doit être terminé si un téléchargement gif a été détecté.

Le troisième fragment de règle est un partenaire du premier fragment de règle :

```
(msg:"JPEG exploit";?flowbits:isset,http.jpeg;content:"|FF|";
pcre:"?/\xFF[\xE1\xE2\xED\xFE]\x00[\x00\x01]/";)
```

Le diagramme suivant illustre l'effet du mot-clé `flowbits` dans le fragment de règle précédent :



Dans le troisième fragment de règle, `flowbits:isset,http.jpeg` détermine que l'état `http.jpeg`, désormais non pertinent, est défini et que `content` et `pcre` correspondent au contenu qui serait malveillant dans un fichier

JPEG, mais pas dans un fichier GIF. Le troisième fragment de règle entraîne un événement de faux positif pour une exploit inexistant dans un fichier JPG.

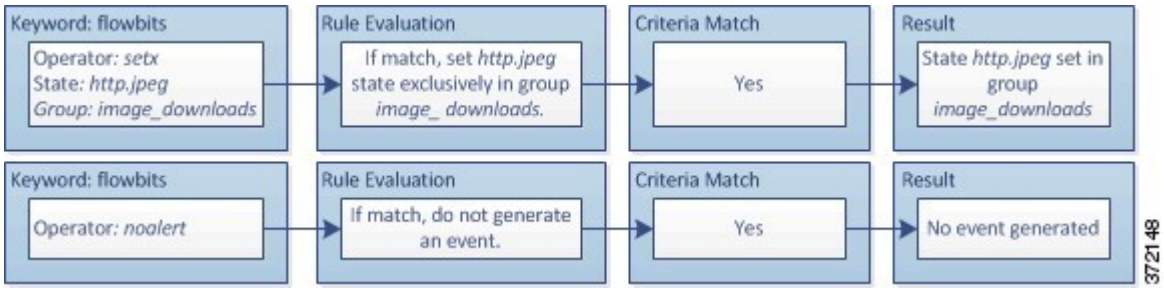
Exemple de mot clé flowbits : configuration pour la protection contre les faux événements positifs

L'exemple suivant montre comment l'inclusion de noms d'états dans un groupe et l'utilisation de l'opérateur `setx` peuvent éviter les faux positifs.

Considérez le même cas que dans l'exemple précédent, sauf que les deux premières règles incluent maintenant leurs deux noms d'état différents dans le même groupe d'états.

```
(msg:"JPEG transfer";
content:"image/";pcre:"/^Content-?Type\x3a(\s*|\s*\r?\n\s+)image\x2fp?jpe?g/smi";
?flowbits:setx,http.jpeg,image_downloads; flowbits:noalert;)
```

Le diagramme suivant illustre l'effet du mot-clé `flowbits` dans le fragment de règle précédent :

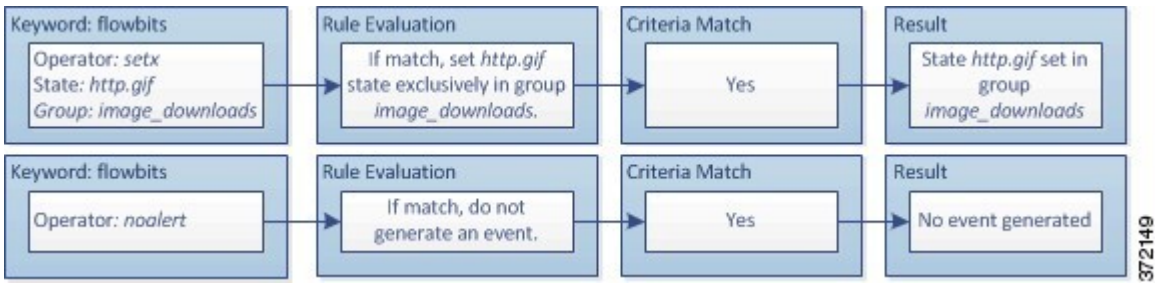


Lorsque le premier fragment de règle détecte un téléchargement de fichier JPEG, le mot-clé `flowbits:setx,http.jpeg,image_downloads` définit l'état de `flowbits` sur `http.jpeg` et inclut l'état dans le groupe `image_downloads`.

La règle suivante détecte ensuite un téléchargement ultérieur de fichier GIF :

```
(msg:"GIF transfer"; content:"image/";
pcre:"/^Content-?Type\x3a(\s*|\s*\r?\n\s+)image\x2fgif/smi";
?flowbits:setx,http.jpg,image_downloads; flowbits:noalert;)
```

Le diagramme suivant illustre l'effet du mot-clé `flowbits` dans le fragment de règle précédent :

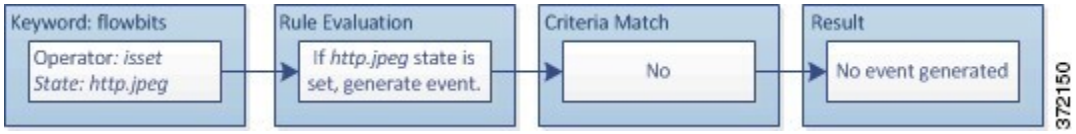


Lorsque le deuxième fragment de règle correspond au téléchargement GIF, le mot-clé `flowbits:setx,http.jpg,image_downloads` définit l'état `http.jpg flowbits` et désactive `http.jpeg`, l'autre état du groupe.

Le troisième fragment de règle ne génère pas de faux positifs :

```
(msg:"JPEG exploit"; ?flowbits:isset,http.jpeg;content:"|FF|";
pcre:"/?\xFF[\xE1\xE2\xED\xFE]\x00[\x00\x01]/";)
```

Le diagramme suivant illustre l'effet du mot-clé `flowbits` dans le fragment de règle précédent :



Comme `flowbits:isset,http.jpeg` a la valeur Faux, le moteur de règles arrête de traiter la règle et aucun événement n'est généré, ce qui évite un faux positif même dans les cas où le contenu du fichier GIF correspond au contenu d'exploitation d'un fichier jpeg.

Le mot-clé http_encode

Vous pouvez utiliser le mot-clé `http_encode` pour générer des événements sur le type de codage dans une requête ou une réponse HTTP avant la normalisation, soit dans l'URI HTTP, dans des données autres que des témoins dans un en-tête HTTP, dans les témoins dans les en-têtes de requêtes HTTP ou set-cookie dans les réponses HTTP.

Vous devez configurer le préprocesseur HTTP Inspect pour inspecter les réponses HTTP et les témoins HTTP pour renvoyer les correspondances pour les règles utilisant le mot-clé `http_encode`.

En outre, vous devez activer l'option de décodage et d'alerte pour chaque type de codage dans la configuration de votre préprocesseur HTTP Inspect afin que le mot-clé `http_encode` dans une règle de prévention des intrusions puisse déclencher des événements sur ce type de codage.

Le tableau suivant décrit les types de codage pour lesquels cette option peut générer des événements dans les URI HTTP, les en-têtes, les témoins et les set-cookies :

Tableau 67 : Types de codage http_encode

Type de codage	Description
utf8	Détecte le codage UTF-8 à l'emplacement spécifié lorsque ce type de codage est activé pour le décodage par le préprocesseur HTTP Inspect.
double_encode	Détecte le double codage à l'emplacement spécifié lorsque ce type de codage est activé pour le décodage par le préprocesseur HTTP Inspect.
non_ascii	Détecte les caractères non-ASCII à l'emplacement spécifié lorsque des caractères non-ASCII sont détectés mais que le type de codage détecté n'est pas activé.
uencode	Détecte l'encodage Microsoft %u à l'emplacement spécifié lorsque ce type de codage est activé pour le décodage par le préprocesseur HTTP Inspect.
bare_byte	Détecte le codage de l'octet nu à l'emplacement spécifié lorsque ce type de codage est activé pour le décodage par le préprocesseur HTTP Inspect.

Sujets connexes

Options de normalisation HTTP au niveau du serveur
Le préprocesseur d'inspection HTTP

Syntaxe du mot-clé http_encode

Emplacement de codage

Spécifie s'il faut rechercher le type de codage précisé dans une URI HTTP, un en-tête ou un témoin, y compris un set-cookie.

Type de codage

Spécifie un ou plusieurs types d'encodage en utilisant l'un des formats suivants :

```
encode_type
encode_type|encode_type|encode_type...
```

où `encode_type` correspond à l'une des valeurs suivantes :

```
utf8
double_encode
non_ascii
uencode
bare_byte.
```

Notez que vous ne pouvez pas utiliser les opérateurs de négation (!) et OR (|) conjointement.

Exemple de mot-clé http_encode : utilisation de deux mots-clés http_endcode pour rechercher deux encodages

L'exemple suivant utilise deux mots-clés `http_encode` dans la même règle pour rechercher l'URI HTTP pour l'encodage UTF-8 ET Microsoft IIS %u :

Tout d'abord, le mot-clé `http_encode` :

- **Emplacement de codage** : `HTTP URI`
- **Type de codage** : `utf8`

Ensuite, le mot-clé `http_encode` supplémentaire :

- **Emplacement de codage** : `HTTP URI`
- **Type de codage** : `uencode`

Présentation : mots-clés file_type et file_group

Les mots-clés `file_type` et `file_group` vous permettent de détecter les fichiers transmis par FTP, HTTP, SMTP, IMAP, POP3 et NetBIOS-ssn (SMB) en fonction de leur type et de leur version. N'utilisez **pas** plus d'un mot-clé `file_type` ou `file_group` dans une même règle de prévention des intrusions.

Astuces La mise à jour de votre base de données de vulnérabilités (VDB) remplit l'éditeur de règles de prévention des intrusions avec les types de fichiers, les versions et les groupes les plus récents.

Remarque Le système n'active pas automatiquement les préprocesseurs de manière à ce qu'ils s'adaptent aux mots-clés `file_type` et `file_group`.

Vous **devez** activer des préprocesseurs spécifiques si vous souhaitez générer des événements et, dans un déploiement en ligne, supprimer les paquets incriminés pour le trafic correspondant aux mots-clés `file_type` ou `file_group`.

Tableau 68 : Génération d'incidents d'intrusion file_type et file_group

Protocole	Option de Préprocesseur requis ou préprocesseur
FTP	Préprocesseur FTP/Telnet et option de normalisation en ligne du préprocesseur **Normaliser la charge utile TCP**
HTTP	HTTP Préprocesseur Inspect pour générer des incidents d'intrusion dans le trafic HTTP
SMTP	Préprocesseur SMTP pour générer des incidents d'intrusion dans le trafic HTTP
IMAP	Préprocesseur IMAP
POP3	Préprocesseur POP
NetBIOS-ssn (SMB)	Le préprocesseur DCE/RPC et l'option de préprocesseur DCE/RPC d'**inspection de fichiers SMB**

Sujets connexes

Le décodeur Telnet/FTP
Le préprocesseur de normalisation en ligne
Le préprocesseur d'inspection HTTP
Le préprocesseur SMTP
Le préprocesseur IMAP
Le préprocesseur POP
Le préprocesseur DCE/RPC

Les mots-clés file_type et file_group

file_type

Le mot-clé `file_type` vous permet de préciser le type de fichier et la version d'un fichier détecté dans le trafic. Les arguments de type de fichier (par exemple, **jpeg** et **PDF**) identifient le format de fichier que vous souhaitez trouver dans le trafic.

Remarque N'utilisez **pas** le mot-clé `file_type` avec un autre mot-clé `file_type` ou `file_group` dans la même règle de prévention des intrusions.

Le système sélectionne **n'importe quelle version** par défaut, mais certains types de fichiers vous permettent de sélectionner des options de version (p. ex., PDF version **1.7**) pour identifier les versions de types de fichiers spécifiques que vous souhaitez trouver dans le trafic.

file_group

Le mot-clé `file_group` vous permet de sélectionner un groupe de types de fichiers similaires défini par Cisco à trouver dans le trafic (par exemple, **multimédia** ou **audio**). Les groupes de fichiers comprennent également les versions définies par Cisco pour chaque type de fichier du groupe.

Remarque N'utilisez **pas** le mot-clé `file_group` avec un autre mot-clé `file_group` ou `file_type` dans la même règle de prévention des intrusions.

Le mot-clé file_data

Le mot-clé `file_data` fournit un pointeur qui sert de référence pour les arguments de position disponibles pour d'autres mots-clés tels que `content`, `byte_jump`, `byte_test` et `pcre`. Le trafic détecté détermine le type de données vers lequel pointe le mot-clé `file_data`. Vous pouvez utiliser le mot-clé `file_data` pour pointer vers le début des types de charge utile suivants :

- Corps de réponse HTTP

  Pour inspecter les paquets de réponse HTTP, le préprocesseur HTTP Inspect doit être activé et vous devez configurer ce dernier pour inspecter les réponses HTTP. Le mot-clé `file_data` correspond si le préprocesseur HTTP Inspect détecte les données du corps de la réponse HTTP.

- Données de fichier gzip non compressées

  Pour inspecter les fichiers gzip non compressés dans le corps de la réponse HTTP, le préprocesseur HTTP Inspect doit être activé et vous devez le configurer pour qu'il inspecte les réponses HTTP et décompresse les fichiers compressés par gzip dans le corps de la réponse HTTP. Pour plus d'informations, voir les options de niveau de la normalisation du serveur HTTP **Inspecter les réponses HTTP** et **Inspecter les données compressées**. Le mot-clé `file_data` correspond si le préprocesseur HTTP Inspect détecte des données gzip non compressées dans le corps de la réponse HTTP.

- JavaScript normalisé

  Pour inspecter des données JavaScript normalisées, le préprocesseur HTTP Inspect doit être activé et vous devez le configurer pour inspecter les réponses HTTP. Le mot-clé `file_data` correspond si le préprocesseur HTTP Inspect détecte JavaScript dans les données du corps de la réponse.

- Charge utile SMTP

  Pour inspecter la charge utile SMTP, le préprocesseur SMTP doit être activé. Le mot-clé `file_data` correspond si le préprocesseur SMTP détecte des données SMTP.

- Pièces jointes codées dans le trafic SMTP, POP ou IMAP

  Pour inspecter les pièces jointes de courriel dans le trafic SMTP, POP ou IMAP, le préprocesseur SMTP, POP ou IMAP, respectivement, doit être activé, seul ou en combinaison avec l'un quelconque des deux autres. Ensuite, pour chaque préprocesseur activé, vous devez vous assurer qu'il est configuré pour décoder chaque type de codage de pièce jointe que vous souhaitez décoder. Les options de décodage de

la pièce jointe que vous pouvez configurer pour chaque préprocesseur sont les suivantes : **Base64 Decoding Depth**, **7-Bit/8-Bit/Binary Decoding Depth**, **Quoted-Printable Decoding Depth**, et **Unix-to-Unix Decoding Depth**.

Vous pouvez utiliser plusieurs mots-clés `file_data` dans une règle.

Sujets connexes

Le préprocesseur d'inspection HTTP
Options de normalisation HTTP au niveau du serveur
Le préprocesseur SMTP
Le préprocesseur IMAP

Le mot-clé pkt_data

Le mot-clé `pkt_data` fournit un pointeur qui sert de référence pour les arguments positionnels disponibles pour d'autres mots-clés tels que `content`, `byte_jump`, `byte_test` et `pcre`.

Lorsqu'un trafic FTP, Telnet ou SMTP normalisé est détecté, le mot-clé `pkt_data` pointe vers le début de la charge utile de paquet normalisé. Lorsqu'un autre trafic est détecté, le mot-clé `pkt_data` pointe vers le début de la charge utile TCP ou UDP brute.

Les options de normalisation suivantes doivent être activées pour que le système normalise le trafic correspondant pour les règles d'inspection par intrusion :

- Activez l'option **Détecter les codes d'échappement Telnet dans les commandes FTP** du préprocesseur FTP et Telnet pour normaliser le trafic FTP pour l'inspection.
- Activez l'option **Normalize** telnet du préprocesseur FTP et Telnet pour normaliser le trafic Telnet aux fins d'inspection.
- Activez l'option **Normalize** du préprocesseur SMTP pour normaliser le trafic SMTP à des fins d'inspection.

Vous pouvez utiliser plusieurs mots-clés `pkt_data` dans une règle.

Sujets connexes

Options FTP au niveau du client
Options Telnet
Options du préprocesseur SMTP

Les mots-clés base64_decode et base64_data

Vous pouvez utiliser les mots-clés `base64_decode` et `base64_data` pour demander au moteur de règles de décoder et d'inspecter les données spécifiées en tant que données Base64. Cela peut être utile, par exemple, pour inspecter les en-têtes de demande d'authentification HTTP codés en Base64 et les données codées en Base64 dans les demandes HTTP PUT et POST.

Ces mots-clés sont particulièrement utiles pour décoder et inspecter les données Base64 dans les requêtes HTTP. Cependant, vous pouvez également les utiliser avec n'importe quel protocole tel que SMTP qui utilise les espaces et les tabulations de la même manière que HTTP utilise ces caractères pour étendre une longue ligne d'en-tête sur plusieurs lignes. Lorsque ce prolongement de ligne, appelé repliement, n'est pas présent dans un protocole qui l'utilise, l'inspection se termine à tout retour à la ligne ou à tout saut de ligne qui n'est pas suivi d'une espace ou d'une tabulation.

base64_decode

Le mot-clé `base64_decode` indique au moteur de règles de décoder les données des paquets en tant que données Base64. Les arguments facultatifs vous permettent de préciser le nombre d'octets à décoder et l'endroit où commencer le décodage des données.

Vous pouvez utiliser le mot-clé `base64_decode` une seule fois dans une règle; il doit précéder au moins une instance du mot-clé `base64_data`.

Avant de décoder les données Base64, le moteur de règles déploie de longs en-têtes qui sont pliés sur plusieurs lignes. Le décodage se termine lorsque le moteur de règles rencontre l'un des événements suivants :

- à la fin d'une ligne d'en-tête
- le nombre d'octets à décoder est atteint
- la fin du paquet

Le tableau suivant décrit les arguments que vous pouvez utiliser avec le mot-clé `base64_decode`.

Tableau 69 : Arguments base64_decode facultatifs

Argument	Description
Octets	Spécifie le nombre d'octets à décoder. Lorsque non spécifié, le décodage se poursuit jusqu'à la fin d'une ligne d'en-tête ou jusqu'à la fin de la charge utile du paquet, selon la première éventualité. Vous pouvez spécifier une valeur positive non nulle.
Décalage	Détermine le décalage par rapport au début de la charge utile du paquet ou, lorsque vous spécifiez également **relative**, par rapport à l'emplacement d'inspection actuel. Vous pouvez spécifier une valeur positive non nulle.
Relatif	Spécifie l'inspection par rapport à l'emplacement d'inspection actuel.

base64_data

Le mot-clé `base64_data` fournit une référence pour l'inspection des données Base64 décodées à l'aide du mot-clé `base64_decode`. Le mot-clé `base64_data` définit que l'inspection commence au début des données Base64 décodées. Vous pouvez ensuite utiliser les arguments positionnels disponibles pour d'autres mots-clés tels que `content` ou `byte_test` afin de préciser l'emplacement à inspecter.

Vous devez utiliser le mot-clé `base64_data` au moins une fois après le mot-clé `base64_decode` ; vous pouvez éventuellement utiliser `base64_data` plusieurs fois pour revenir au début des données Base64 décodées.

Tenez compte des éléments suivants lors de l'inspection de données en base64 :

- Vous ne pouvez pas utiliser l'outil de recherche de modèle rapide.
- Si vous interrompez l'inspection Base64 dans une règle entre deux arguments de contenu HTTP, vous devez insérer un autre mot-clé `base64_data` dans la règle avant d'inspecter plus avant les données Base64.

Sujets connexes

Présentation : Contenu HTTP et arguments du mot-clé protected_content, à la page 30
Arguments de la recherche de schéma rapide pour mot-clé de contenu, à la page 35

À propos de la traduction

Cisco peut fournir des traductions du présent contenu dans la langue locale pour certains endroits. Veuillez noter que des traductions sont fournies à titre informatif seulement et, en cas d’incohérence, la version anglaise du présent contenu prévaudra.