Configuración de AppID Early Packet Detection en Secure Firewall Threat Defence 7.4

Contenido

Introducción

Este documento describe cómo configurar AppID Early Packet Detection en Cisco Secure Firewall 7.4.

Antecedentes - Problema (requisitos del cliente)

- La detección de aplicaciones mediante la inspección profunda de paquetes puede requerir más de un paquete para identificar el tráfico.
- A veces, cuando se conoce la IP y/o el puerto de un servidor de aplicaciones, puede evitar inspeccionar paquetes adicionales.

Qué hay de nuevo

- Se ha creado una nueva API Lua AppID basada en Snort que nos permite asignar una dirección IP, un puerto y un protocolo a la dirección respectiva:
  - Protocolo de aplicación (servicio appid),
  - Aplicación cliente (cliente appid) y
  - Aplicación web (payload appid).
- Se pueden crear detectores de aplicaciones personalizados en FMC mediante esta API para la detección de aplicaciones.
- Una vez activado este detector, esta nueva API nos permitiría identificar aplicaciones en el primer paquete de una sesión.

Descripción general de características

- La API se identifica como:
  - **addHostFirstPktApp** (protocol_appId, client_appId, payload_appId, dirección IP, puerto, protocolo, volver a inspeccionar)

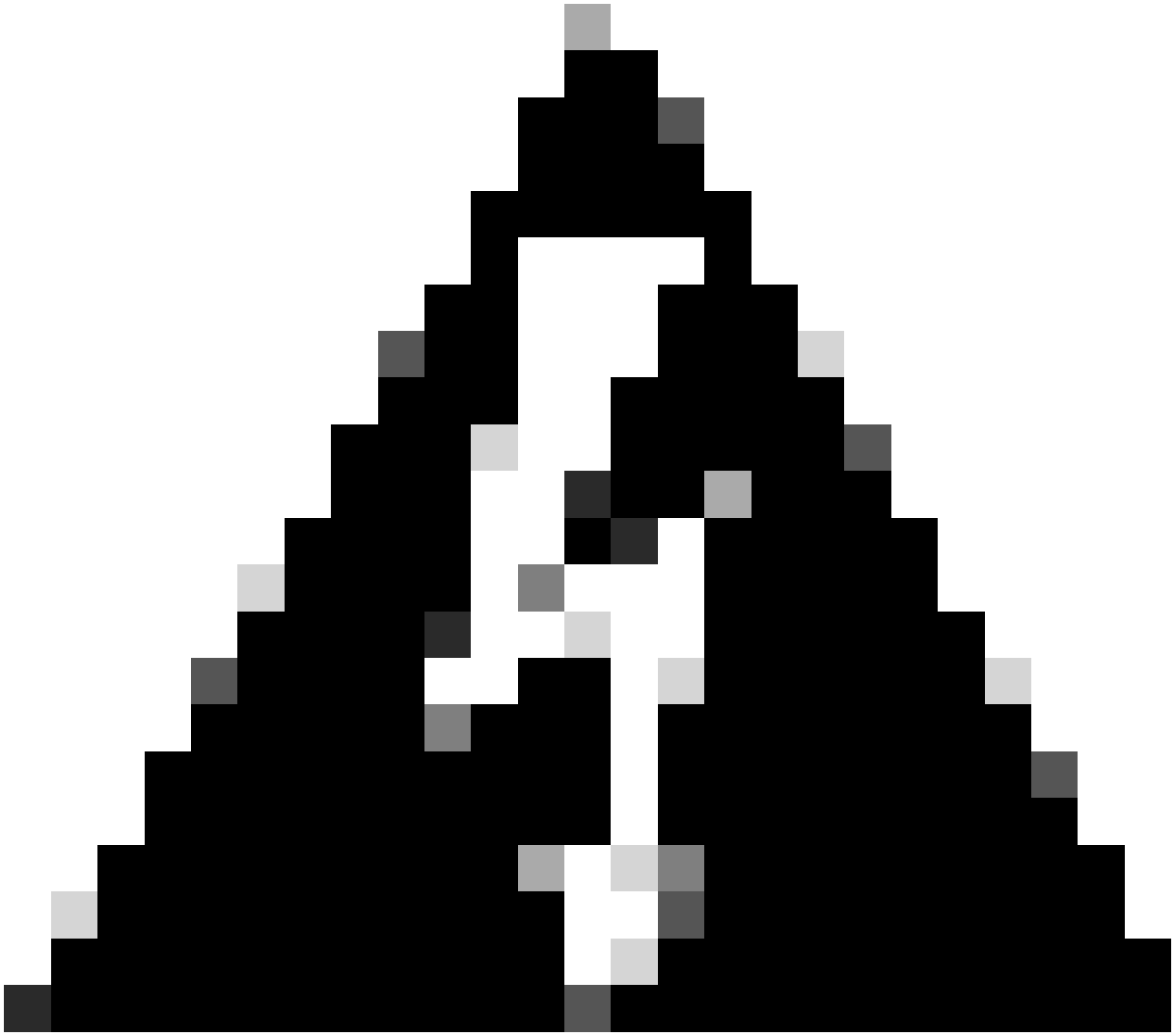
- Se crea una entrada de caché para cada asignación creada en el detector de aplicaciones personalizado.
- El primer paquete de todas las sesiones entrantes se inspecciona para ver si se encuentra una coincidencia en la caché.
- Una vez que se encuentra una coincidencia, asignamos los appids correspondientes para la sesión y el proceso de detección de aplicaciones se detiene.
- Los usuarios tienen la opción de volver a inspeccionar el tráfico incluso después de que la API haya encontrado una coincidencia.
- El argumento reinspect es un valor booleano que indica si hay necesidad de reexaminar las aplicaciones encontradas en el primer paquete o no.
- Cuando la reinspección es verdadera, la detección de aplicaciones continúa incluso si la API encuentra una coincidencia.
- En este caso, los appids asignados en el primer paquete pueden cambiar.

Prerrequisitos, Plataformas Soportadas, Licencias

Plataformas mínimas de software y hardware

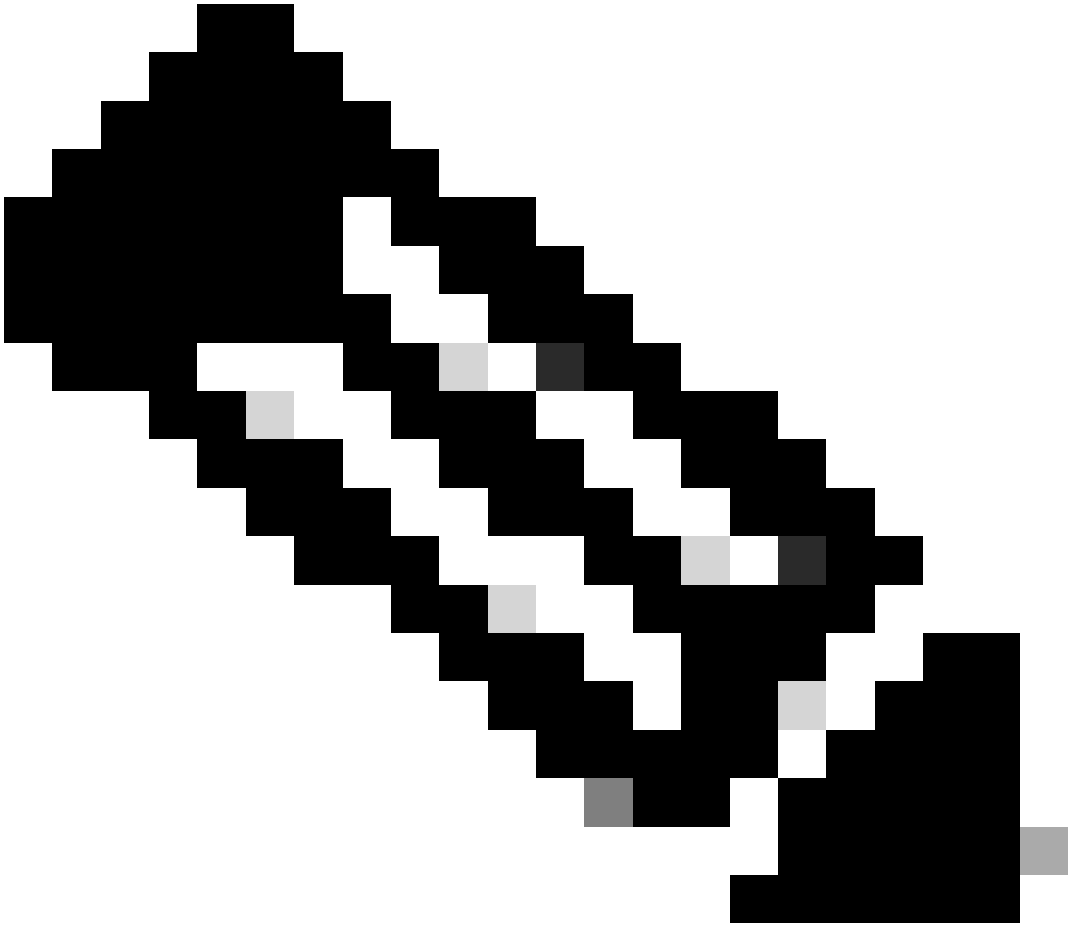
Aplicación y versión mínima	Plataformas gestionadas y versiones admitidas	Gerente(s)	Notas
Firewall seguro 7.4	Todas las	FMC en las	Esta es una función

Uso de Snort3	plataformas compatibles con FTD 7.4	instalaciones + FTD	del dispositivo; FTD debe estar en la versión 7.4



Advertencia: Snort 2 no admite esta API.

Compatibilidad con Snort 3, multiinstancia y HA/agrupación en clústeres



Nota: Requiere que el motor de detección sea Snort 3.

FTD	
¿Se admiten varias instancias?	Yes
Compatible con dispositivos HA.	Yes
¿Es compatible con dispositivos	Yes

agrupados?	

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.
· Cisco Firepower Threat Defence con versión 7.4 o superior.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Detalles de la función

Descripción de la función funcional

Comparación con versiones anteriores a esta versión

En Secure Firewall 7.3 y versiones anteriores	Novedad para Secure Firewall 7.4
· La detección de aplicaciones para una combinación conocida de IP/puerto/protocolo solo estaba disponible como opción alternativa después de agotar todos los demás mecanismos de detección de aplicaciones. · Básicamente, no se admitía la detección en el primer paquete de una sesión.	· La nueva API del detector de lua se evalúa antes que cualquier otro mecanismo de detección de aplicaciones, · Por lo tanto, en la versión 7.4, admitimos la detección en el primer paquete de una sesión.

Cómo funciona

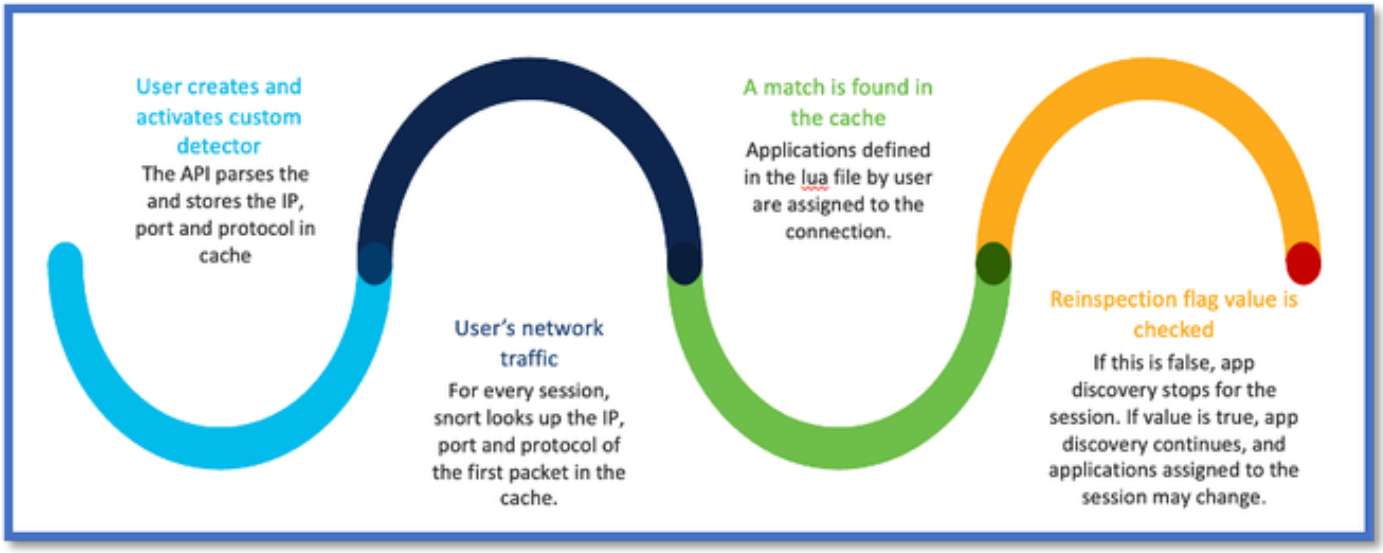
- Crear un archivo lua: asegúrese de que el archivo está en la plantilla lua (sin errores de sintaxis). También verifique que los argumentos dados a la API en el archivo sean correctos.

- Crear un nuevo detector personalizado: Crear un nuevo detector personalizado en FMC y cargar su archivo lua en él. Activa el detector.

- Tráfico de ejecución: envíe al dispositivo el tráfico que coincida con la combinación de IP/puerto/protocolo definida en el detector de aplicaciones personalizadas.

- Comprobar eventos de conexión: en FMC, compruebe los eventos de conexión filtrados por la IP y el puerto. Se identificarían las aplicaciones definidas por el usuario.

Flujo de trabajo API de AppID Early Packet Detection



Ejemplo de Descripción de Campos API desde Detector Personalizado

gDetector:addHostFirstPktApp

(gAppIdProto, gAppIdClient, gAppId, 0, "192.0.2.1", 443, DC.ipproto.tcp);

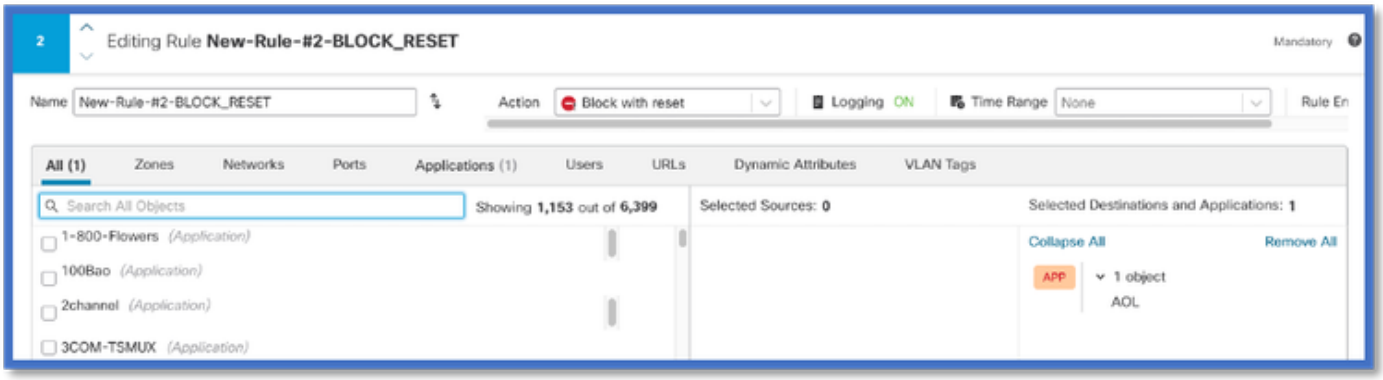
- Los argumentos resaltados son los valores definidos por el usuario para el indicador de reinspección, la dirección IP, el puerto y el protocolo.
- 0 indica un comodín.

Argumentos	Explicación	Valores esperados
Indicador de nueva inspección	Si un usuario prefiere inspeccionar el tráfico en lugar de realizar una acción de firewall basada en IP/Puerto/Protocolo, puede habilitar el valor del indicador de reinspección en 1.	0 = volver a inspeccionar desactivado o 1 = reinspección habilitada

IP Address	IP de destino (única o intervalo de IP en una subred) del servidor. IP de destino del 1er paquete en una sesión.	192.168.4.198 O 192.168.4.198/24 O 2a03:280:f103:83:face:b00c:0:25de O 2a03:280:f103:83:face:b00c:0:25de/32
Puerto	Puerto de destino del 1er paquete en una sesión.	0 a 65535
Protocolo	Protocolo de red	TCP/UDP/ICMP

Caso práctico: Cómo bloquear el tráfico más rápidamente

- Vista de políticas: regla de bloqueo para la aplicación "AOL".



- Prueba del tráfico mediante curl con: curl https://www.example.com v/s curl https://192.0.2.1/ (una de las direcciones IP de TEST)

<#root>

```
> curl https://www.example.com/
```

```
curl: (35) OpenSSL SSL_connect: SSL_ERROR_SYSCALL in connection to www.example.com:443
```

```
> curl https://192.0.2.1/
```

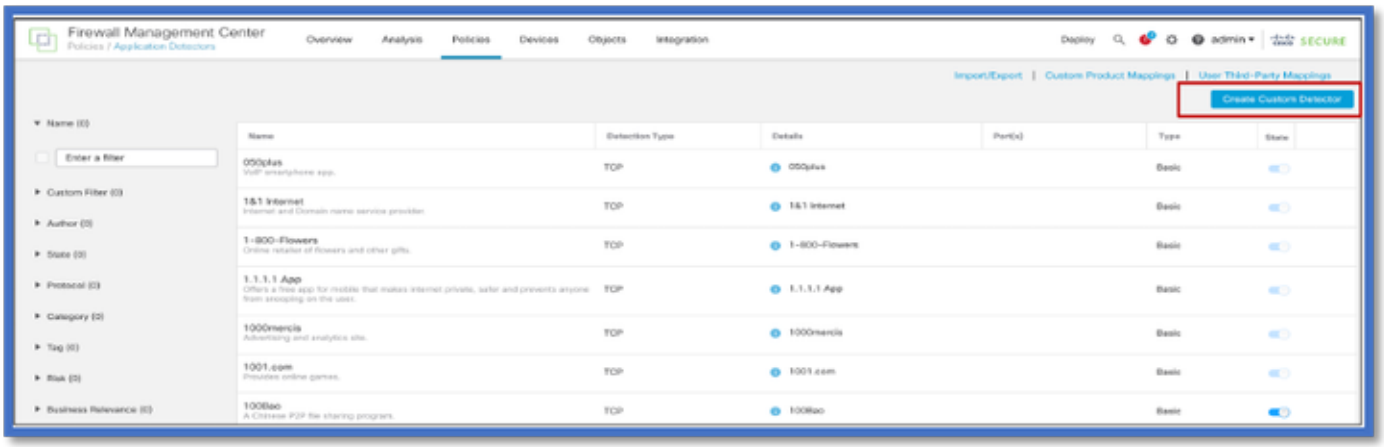
```
curl: (7) Failed to connect to 192.0.2.1 port 443: Connection refused
```

Firewall Management Center (Tutorial)

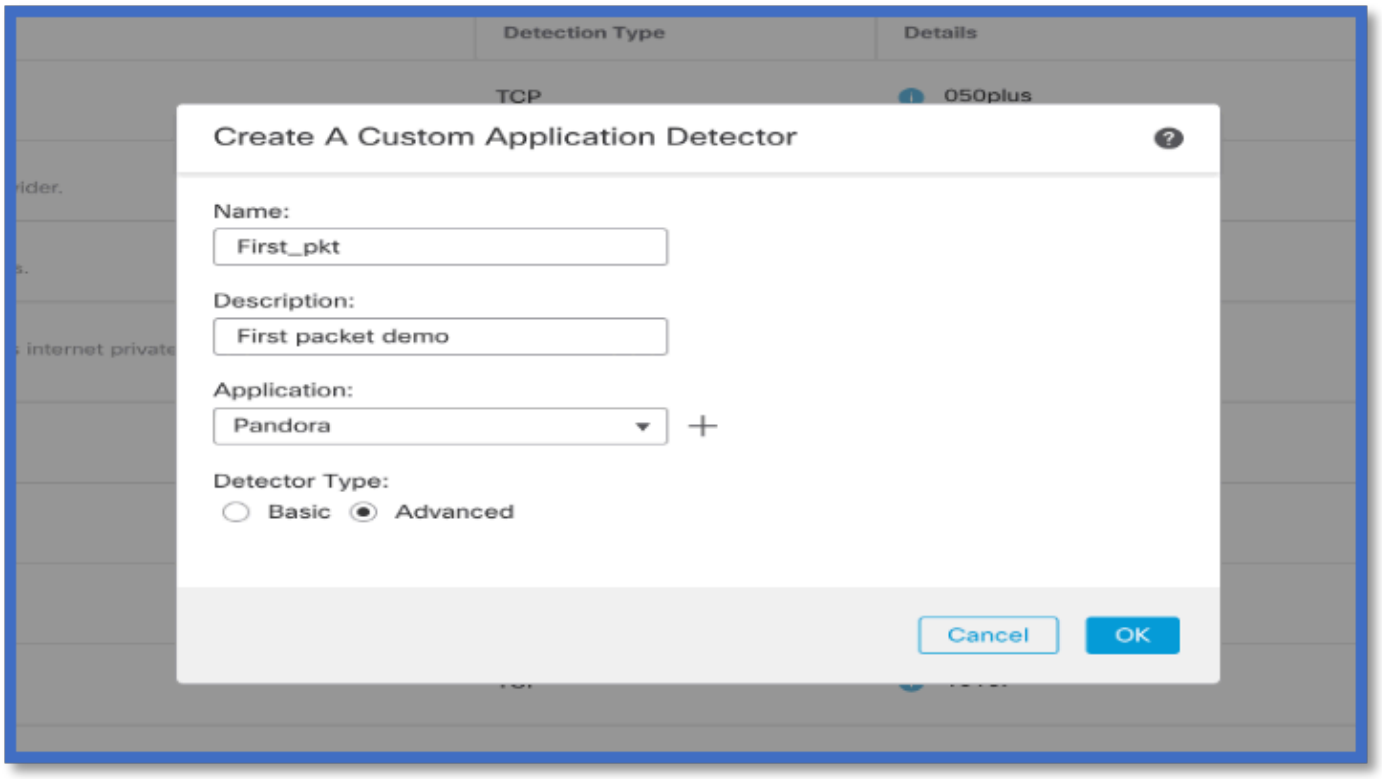
Pasos para crear un detector personalizado mediante la API

Cree un nuevo detector personalizado en el FMC a partir de:

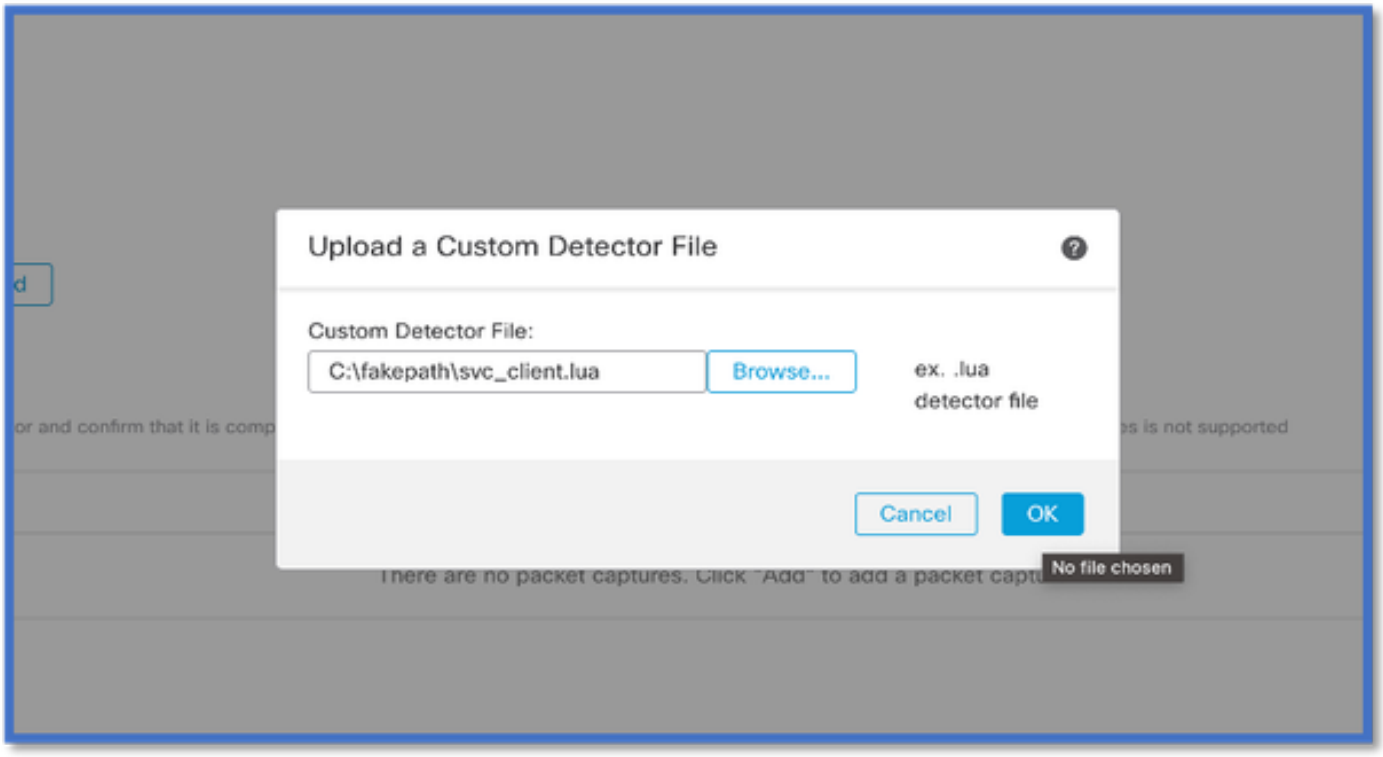
- Policies > Application Detectors > Create Custom Detector .



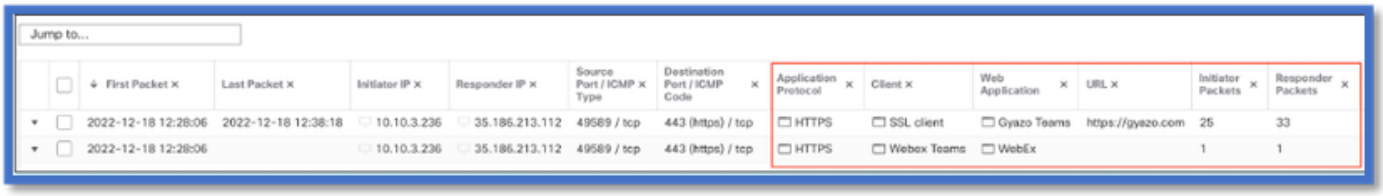
- Definir nombre y descripción.
    - Seleccione la aplicación en el menú desplegable.
    - Seleccione Tipo de detector avanzado.



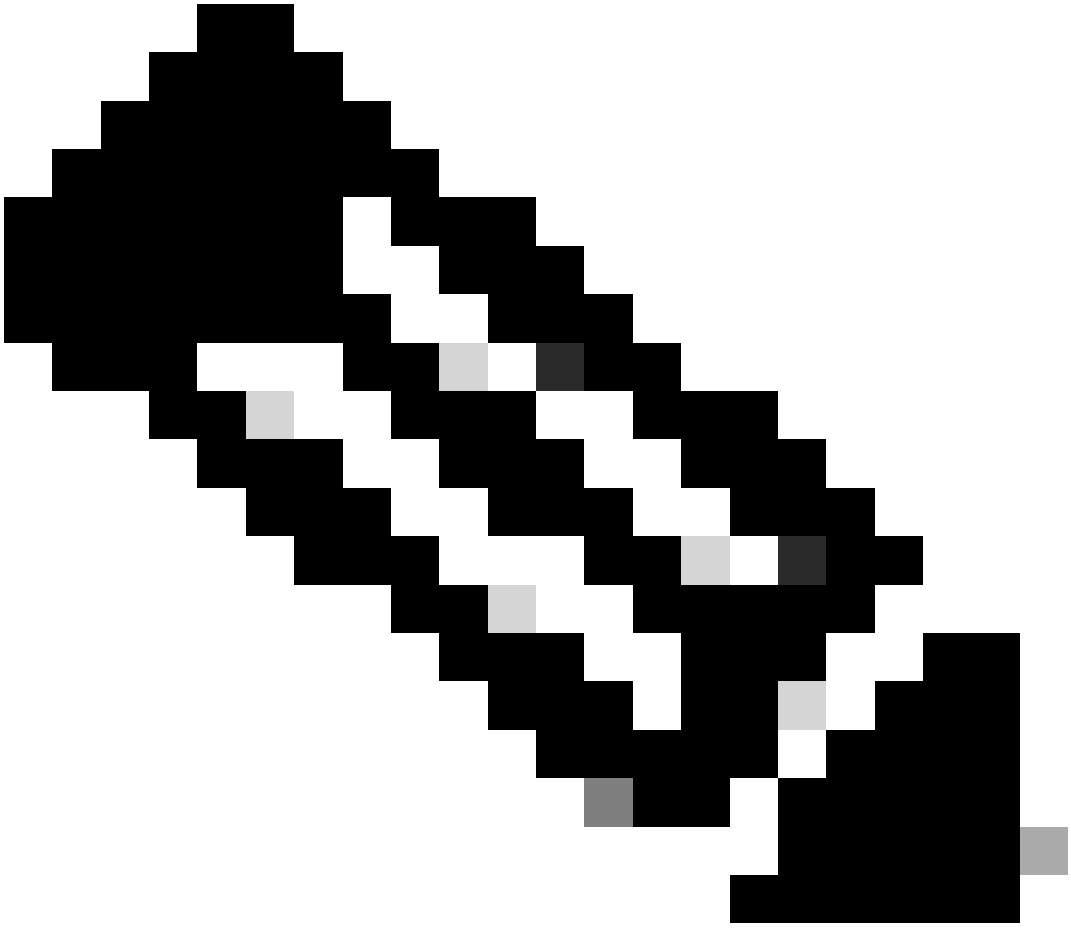
- Cargue el archivo Lua en Criterios de detección. Guarde y active el detector.



Volver a inspeccionar las v/s habilitadas desactivadas



- Los dos eventos muestran el inicio de la conexión v/s el final de la conexión cuando se habilita la nueva inspección.



Nota: Aspectos a destacar:

1. Los "equipos HTTPS, Webex y Webex" se identifican mediante la API al principio de la conexión. Dado que la reinspección es verdadera, el descubrimiento de aplicaciones continúa y los ID de aplicaciones se actualizan a ‘HTTPS, SSL Client y Gyazo Teams’.

2. Observe el número de paquetes de iniciador y de respuesta. Los métodos de detección de aplicaciones regulares requieren muchos más paquetes que la API.

Resolución de problemas/Diagnóstico

Descripción general del diagnóstico

- Se agregan nuevos registros en la depuración de identificación de aplicaciones de soporte del sistema para indicar si la API de detección de primer paquete encuentra alguna aplicación.
- Los registros también muestran si el usuario eligió volver a inspeccionar el tráfico.
- El contenido del archivo del detector de lua cargado por el usuario se puede encontrar en el FTD en /var/sf/appid/custom/lua/<UUID> .
- Cualquier error en el archivo lua se vuelca en el FTD en el archivo /var/log/messages en el momento de activar el detector.

CLI: compatibilidad del sistema application-identification-debug

<#root>

192.0.2.1 443 -> 192.168.1.16 51251 6 AS=4 ID=0 New AppId session

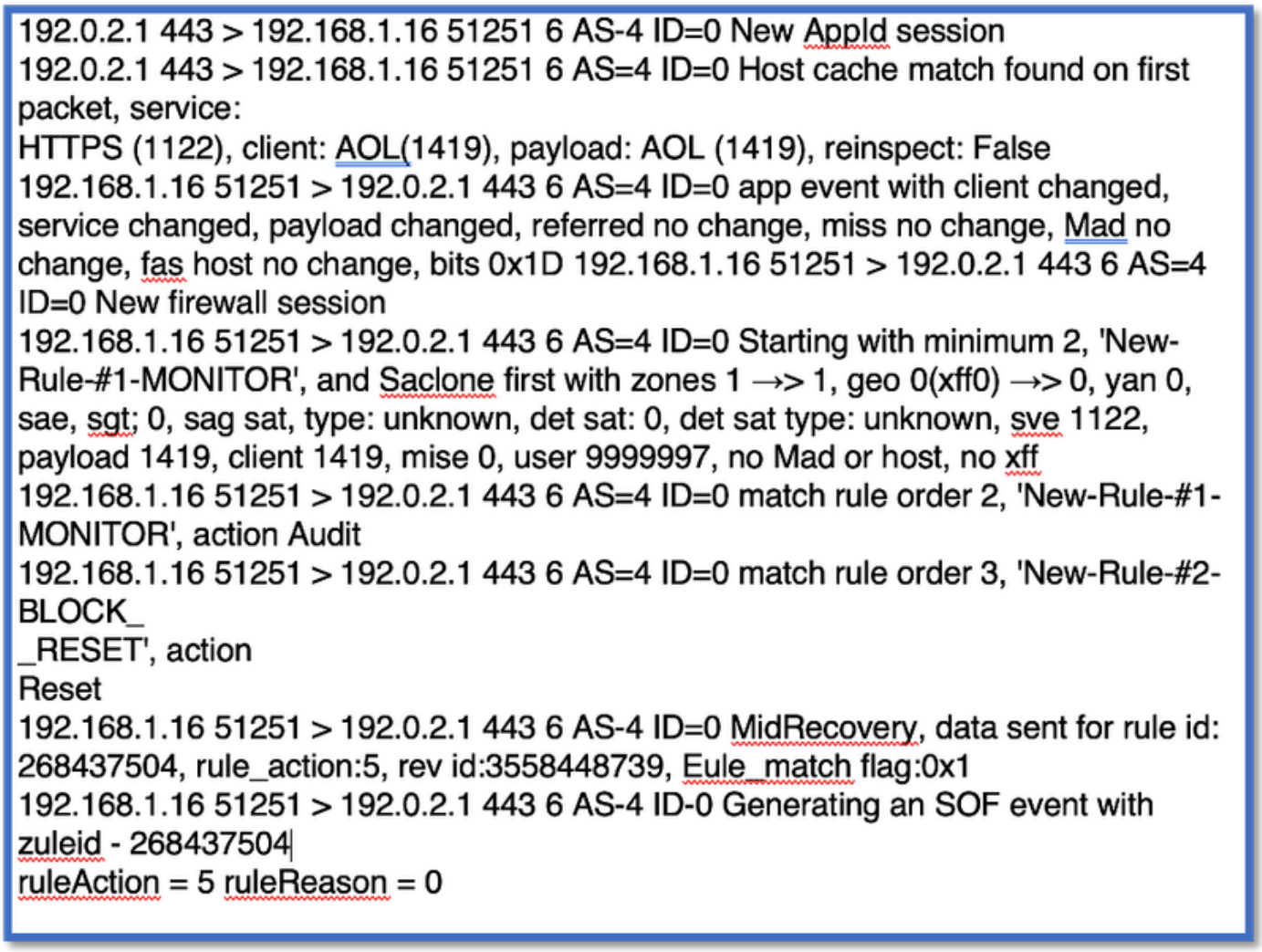
192.0.2.1 443 -> 192.168.1.16 51251 6 AS=4 ID=0 Host cache match found on first packet, service: HTTPS(1

192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 app event with client changed, service changed, payload
192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 New firewall session
192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 Starting with minimum 2, 'New-Rule-#1-MONITOR', and Src
192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 match rule order 2, 'New-Rule-#1-MONITOR', action Audit

192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 match rule order 3, 'New-Rule-#2-BLOCK_RESET', action Re

192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 MidRecovery data sent for rule id: 268437504, rule_acti
192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 Generating an SOF event with rule_id = 268437504 ruleAc

192.168.1.16 51251 -> 192.0.2.1 443 6 AS=4 ID=0 reset action



Ubicación del contenido de los detectores Lua de AppID

Para confirmar si el detector Lua con esta nueva API existe en el dispositivo/FTD, puede ver si la API addHostFirstPktApp se está utilizando en las 2 carpetas de detector de aplicaciones:

1. Detectores VDB AppID -/var/sf/appid/odp/lua

2. Detectores personalizados -/var/sf/appid/custom/lua

Por ejemplo:grep addHostFirstPktApp * en cada carpeta.

Problemas de muestra:

- Problema: el detector Lua personalizado no está activado en el FMC.

Ubicación para comprobar: /var/sf/appid/custom/lua/

Resultado esperado: debe existir un archivo por cada detector de aplicaciones personalizado activado en el FMC. Verifique que el contenido coincida con el archivo lua cargado.

- Problema: el archivo del detector de lua cargado tiene errores.

Archivo para comprobar: /var/log/messages on FTD

Registro de errores:

<#root>

Dec 18 14:17:49 intel-x86-64 SF-IMS[15741]:

Error - appid: can not set env of Lua detector /ngfw/var/sf/appid/custom/lua/6698fbd6-7ede-11ed-972c-d12

Pasos para la resolución de problemas

Problema: aplicaciones no identificadas correctamente para el tráfico que va a la dirección IP y al puerto definidos por el usuario.

Pasos para Solucionar Problemas:

- Verifique que el detector lua esté correctamente definido y activado en el FTD.
  - Verifique el contenido del archivo lua en el FTD y verifique que no se vean errores al activar.
- Verifique la IP de destino, el puerto y el protocolo del primer paquete en la sesión de tráfico.
  - Puede coincidir con los valores definidos en el detector lua.
- Verifique el system-support-application-identification-debug.

- Busque la línea Host cache match found on first packet. Si falta, indica que la API no ha encontrado ninguna coincidencia.

Limitaciones, detalles, problemas comunes y soluciones

En la versión 7.4, no hay ninguna interfaz de usuario para utilizar la API. El soporte de IU se añadiría en futuras versiones.

Historial de revisión

Revisión:	Fecha de publicación	Comentarios
1.0	18 de julio de 2024	Versión inicial

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).