Configurar a Detecção Antecipada de Pacotes do AppID no Secure Firewall Threat Defense 7.4

Contents

Introdução

Este documento descreve como configurar a Detecção Antecipada de Pacotes AppID no Cisco Secure Firewall 7.4.

Histórico - Problema (Requisitos do Cliente)

- A detecção de aplicativos por meio da Inspeção Profunda de Pacotes pode levar mais de um pacote para identificar o tráfego.
- Às vezes, onde o IP e/ou a porta de um servidor de aplicativos é conhecida, você pode evitar a inspeção de pacotes adicionais.

O que há de novo

- Uma nova API Lua AppID baseada em Snort foi criada, o que nos permite mapear um endereço IP, porta e protocolo para o respectivo:
  - Protocolo de aplicação (service appid),
  - Aplicativo cliente (client appid) e
  - Aplicativo Web (payload appid).
- Os Detectores de Aplicativos Personalizados podem ser criados no FMC usando essa API para detecção de aplicativos.
- Uma vez que esse detector é ativado, essa nova API nos permite identificar aplicativos no primeiro pacote em uma sessão.

Visão geral do recurso

- A API é identificada como:
  - **addHostFirstPktApp** (protocol_appId, client_appId, payload_appId, endereço IP, porta, protocolo, inspecionar novamente)

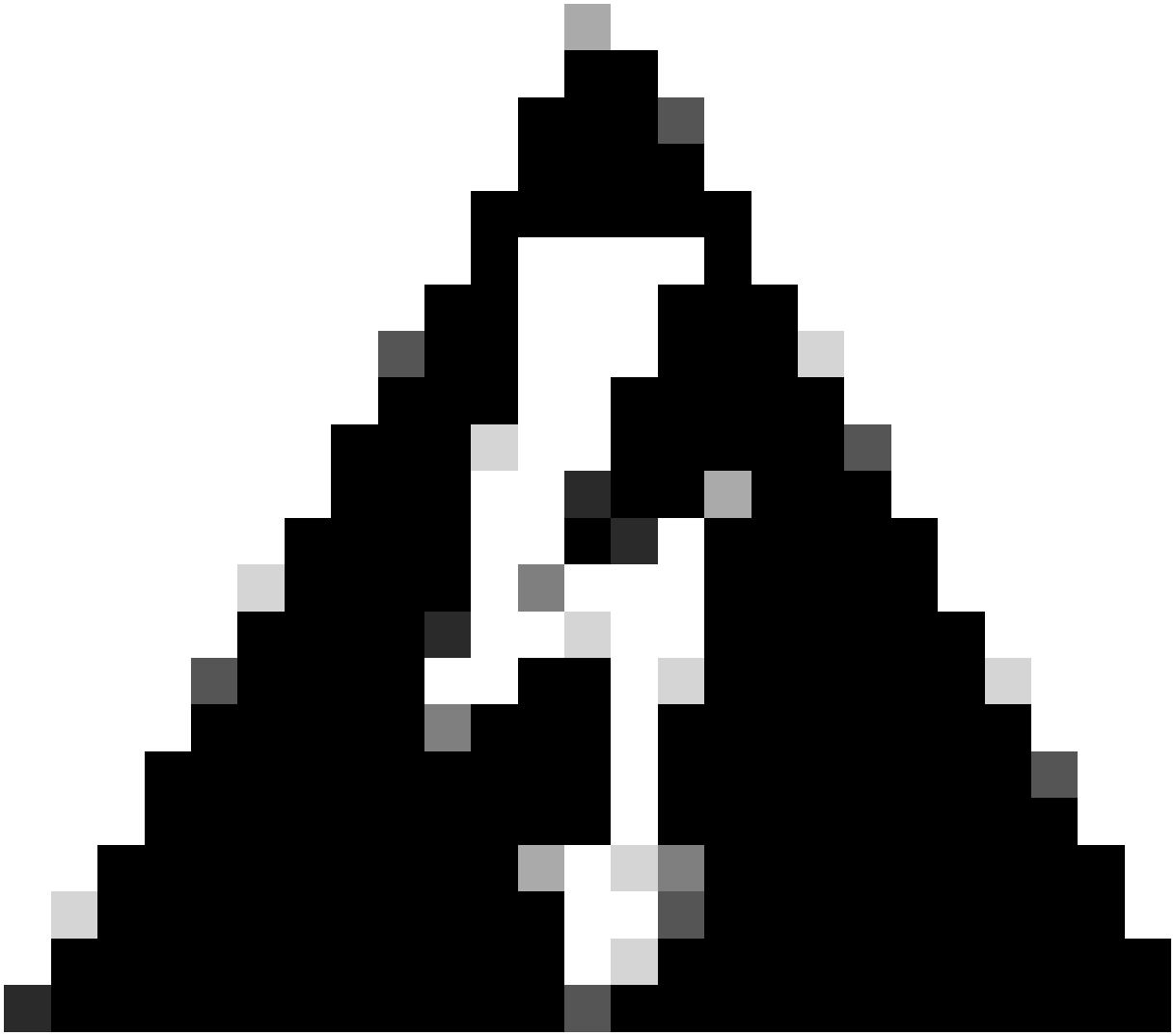
- Uma entrada de cache é criada para cada mapeamento criado no detector de aplicativo personalizado.
- O primeiro pacote de todas as sessões de entrada é inspecionado para ver se uma correspondência é encontrada no cache.
- Quando uma correspondência é encontrada, atribuímos os aplicativos correspondentes para a sessão e o processo de descoberta de aplicativos é interrompido.
- Os usuários têm a opção de inspecionar novamente o tráfego mesmo depois que uma correspondência for encontrada pela API.
- O argumento reinspect é um valor booliano que indica se há necessidade de inspecionar novamente os aplicativos encontrados no primeiro pacote ou não.
- Quando a reinspecção é verdadeira, a descoberta de aplicativos continua mesmo se a API encontrar uma correspondência.
- Nesse caso, os apids atribuídos no primeiro pacote podem ser alterados.

Pré-requisitos, plataformas suportadas, licenciamento

Plataformas mínimas de software e hardware

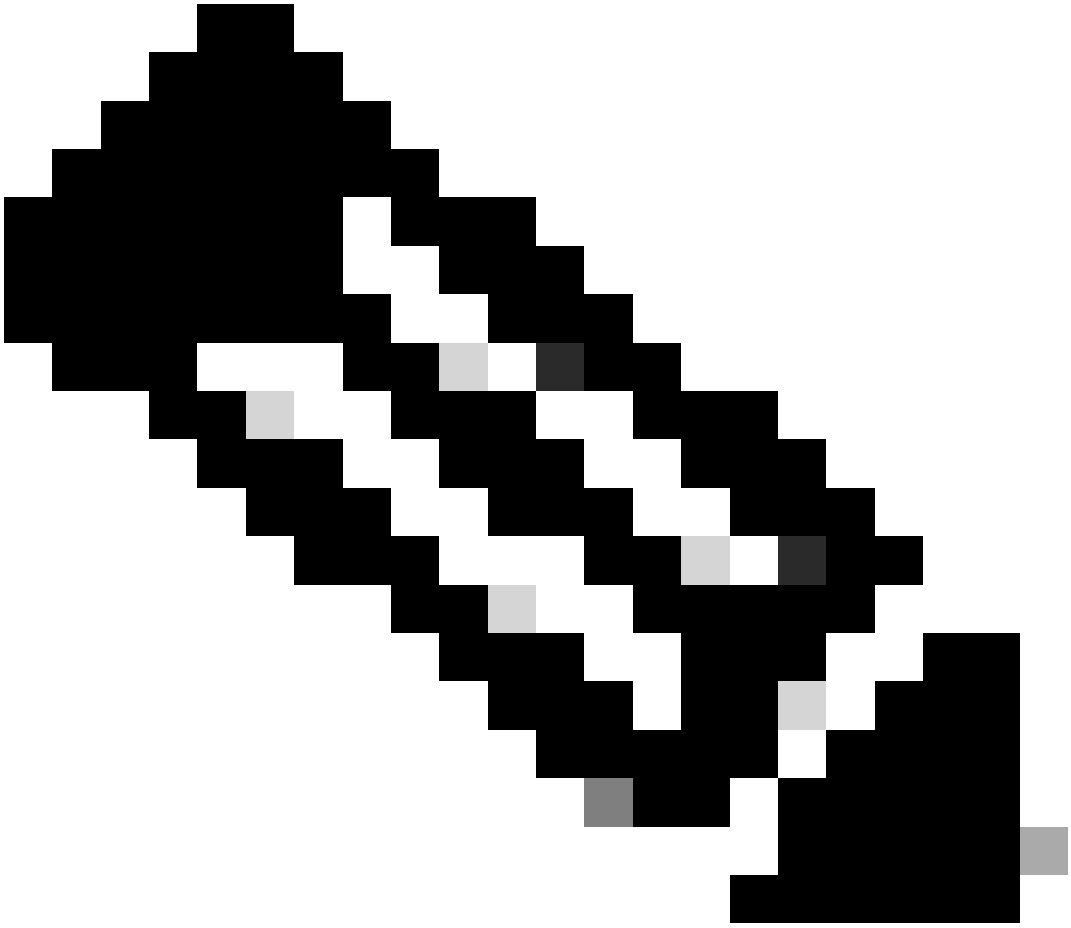
Aplicativo e Versão Mínima	Plataforma(s) gerenciada(s) e versão compatíveis	Gerente(s)	Notas
Firewall seguro 7.4	Todas as	FMC Local + FTD	Este é um recurso

Uso do Snort3	plataformas que oferecem suporte ao FTD 7.4		do lado do dispositivo; o FTD deve estar em 7.4



Aviso: o Snort 2 não oferece suporte a esta API.

Suporte a Snort 3, várias instâncias e HA/clustering



Observação: exige que o Snort 3 seja o mecanismo de detecção.

FTD	
Suporte a várias instâncias?	Yes
Compatível com dispositivos de alta disponibilidade	Yes

Compatível com dispositivos em cluster?	Yes

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:
· Cisco Firepower Threat Defense executando a versão 7.4 ou posterior.

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

Detalhes do recurso

Descrição do recurso funcional

Comparando com anterior a esta versão

No Secure Firewall 7.3 e inferior	Novidade do Secure Firewall 7.4
· A detecção de aplicativos para uma combinação conhecida de IP/Porta/Protocolo estava disponível apenas como uma opção de retorno após a exaustão de todos os outros mecanismos de detecção de aplicativos. · Essencialmente, a detecção no primeiro pacote em uma sessão não era suportada.	· A nova API de detector de lua é avaliada antes de qualquer outro mecanismo de detecção de aplicativo, · Assim, na versão 7.4, suportamos a detecção no primeiro pacote de uma sessão.

Como funciona

- Crie um arquivo lua: verifique se o arquivo está no modelo lua (sem erros de sintaxe). Verifique também se os argumentos fornecidos à API no arquivo estão corretos.

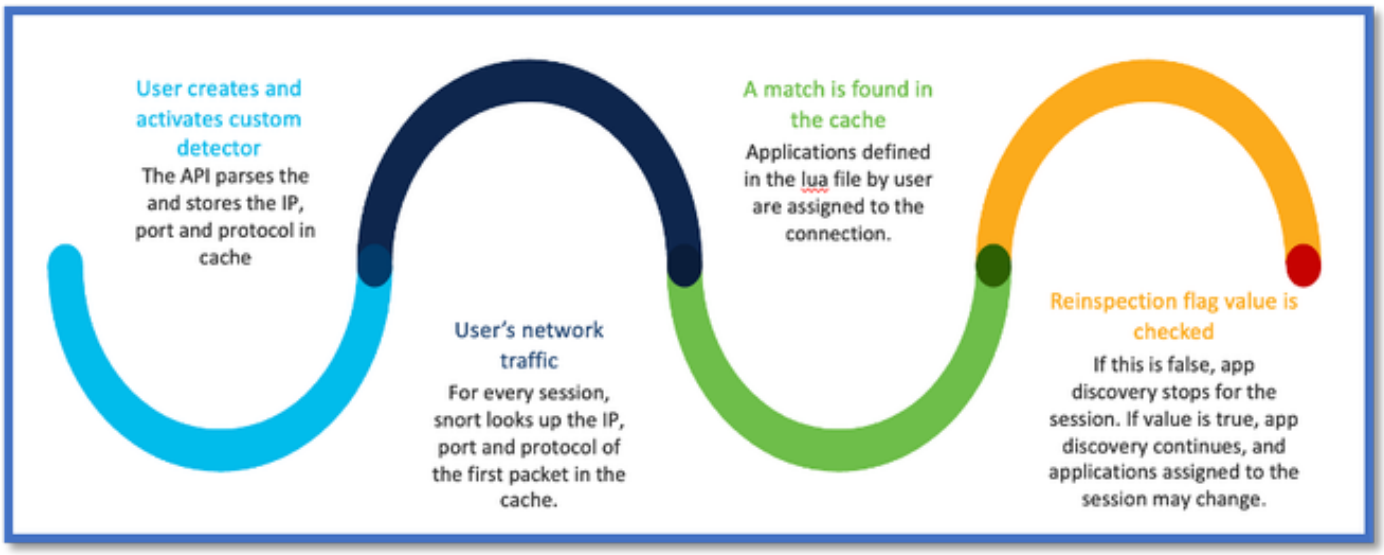
- Crie um novo detector personalizado: crie um novo detector personalizado no FMC e carregue seu arquivo lua nele. Ative o detector.

- Tráfego de execução: envia o tráfego que corresponde à combinação de IP/porta/protocolo definida no detector de aplicativo

personalizado para o dispositivo.

- Verificar eventos de conexão: no FMC, verifique os eventos de conexão filtrados pelo IP e pela porta. Os aplicativos definidos pelo usuário seriam identificados.

Fluxo de Trabalho da API de Detecção Antecipada de Pacote AppID



Descrição dos campos de API a partir do exemplo do detector personalizado

gDetector:addHostFirstPktApp

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

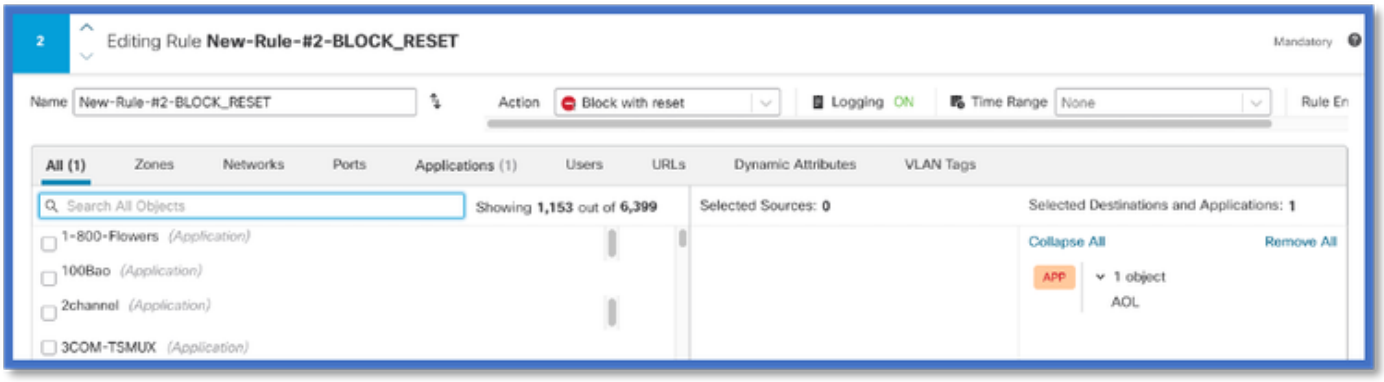
- Os argumentos destacados são os valores definidos pelo usuário para o sinalizador de reinspeção, endereço IP, porta e protocolo.
- 0 indica um curinga.

Argumentos	Explicação	Valores esperados
Sinalizador Reinspect	Se um usuário preferir inspecionar o tráfego em vez de executar uma ação de firewall com base em IP/Porta/Protocolo, ele poderá habilitar o valor do sinalizador de reinspeção para 1.	0 = reinspect desativado ou 1 = reinspect habilitado

IP Address	IP de destino (único ou intervalo de IPs em uma sub-rede) do servidor. IP de destino do 1° pacote em uma sessão.	192.168.4.198 OU 192.168.4.198/24 OU 2a03:2880:f103:83:face:b00c:0:25de OU 2a03:280:f103:83:face:b00c:0:25de/32
Porta	Porta de destino do 1° pacote em uma sessão.	0 a 65535
Protocolo	Protocolo de rede	TCP/UDP/ICMP

Caso de uso: como bloquear o tráfego mais rapidamente

- Exibição de política: Regra de bloqueio para o aplicativo "AOL".



- Testando o tráfego usando curl com: curl https://www.example.com v/s curl https://192.0.2.1/ (um dos endereços IP do 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/
```

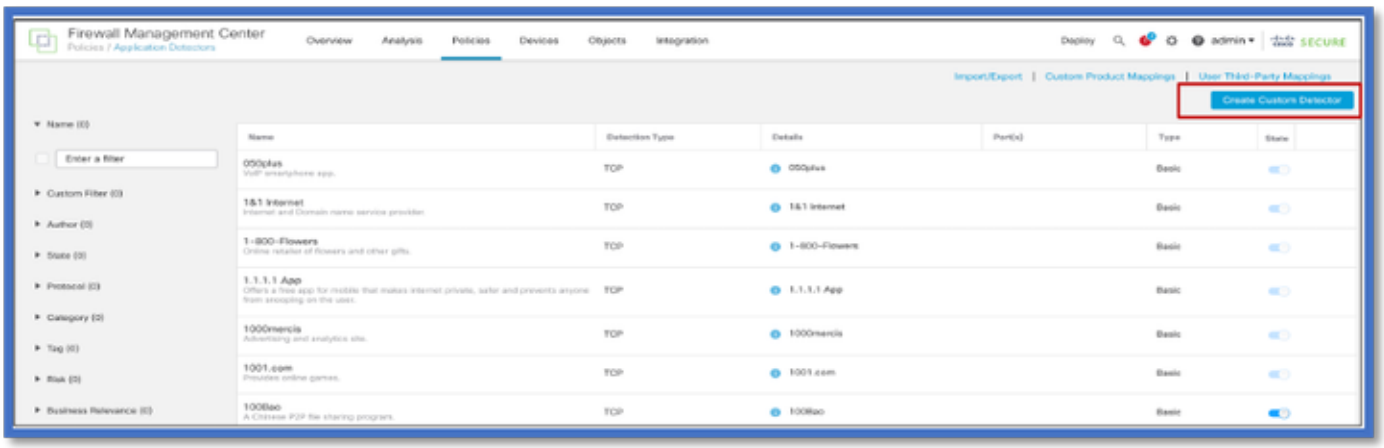
```
curl: (7) Failed to connect to 192.0.2.1 port 443: Connection refused
```

Passo a passo do Firewall Management Center

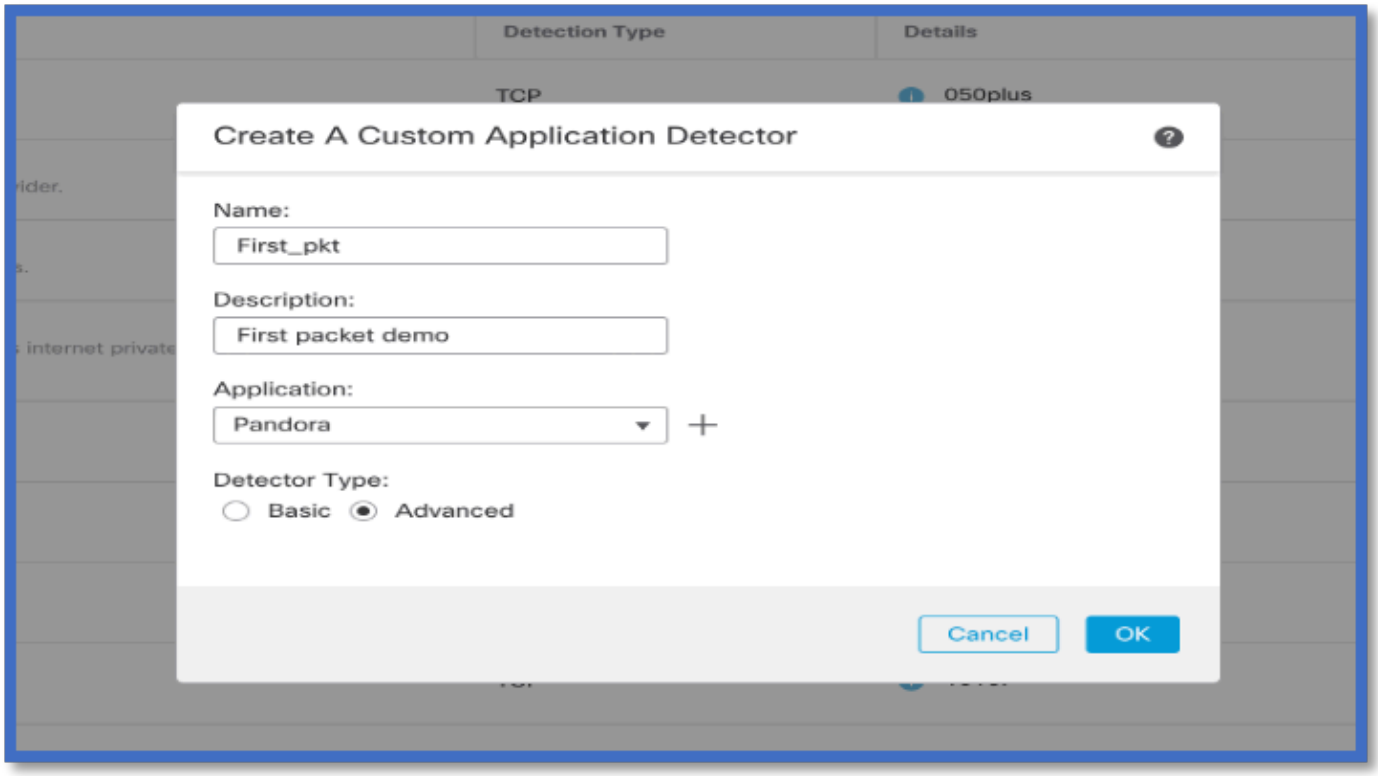
Etapas para criar um detector personalizado usando a API

Criar um novo detector personalizado no FMC de:

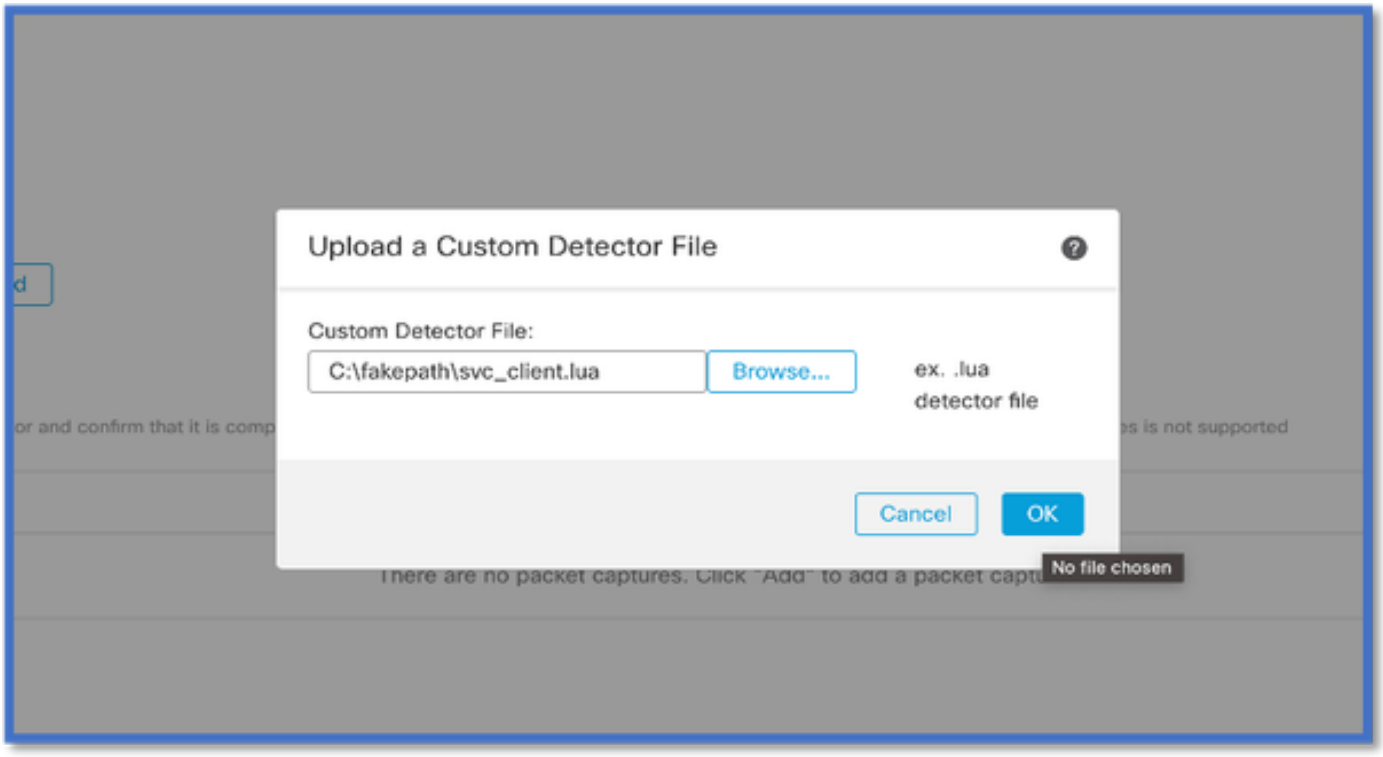
- Policies > Application Detectors > Create Custom Detector .



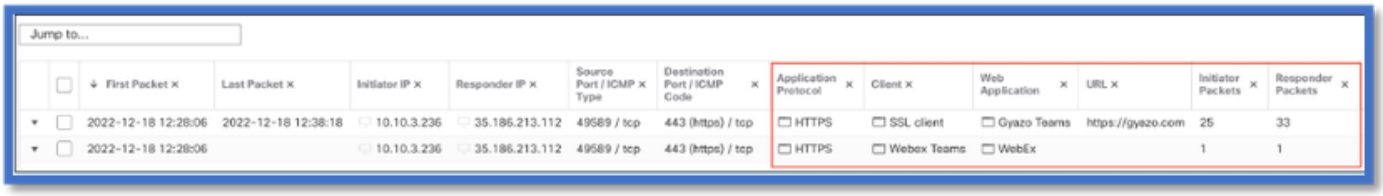
- Definir nome e descrição.
  - Escolha o aplicativo no menu suspenso.
  - Selecione Advanced Detector Type (Tipo de detector avançado).



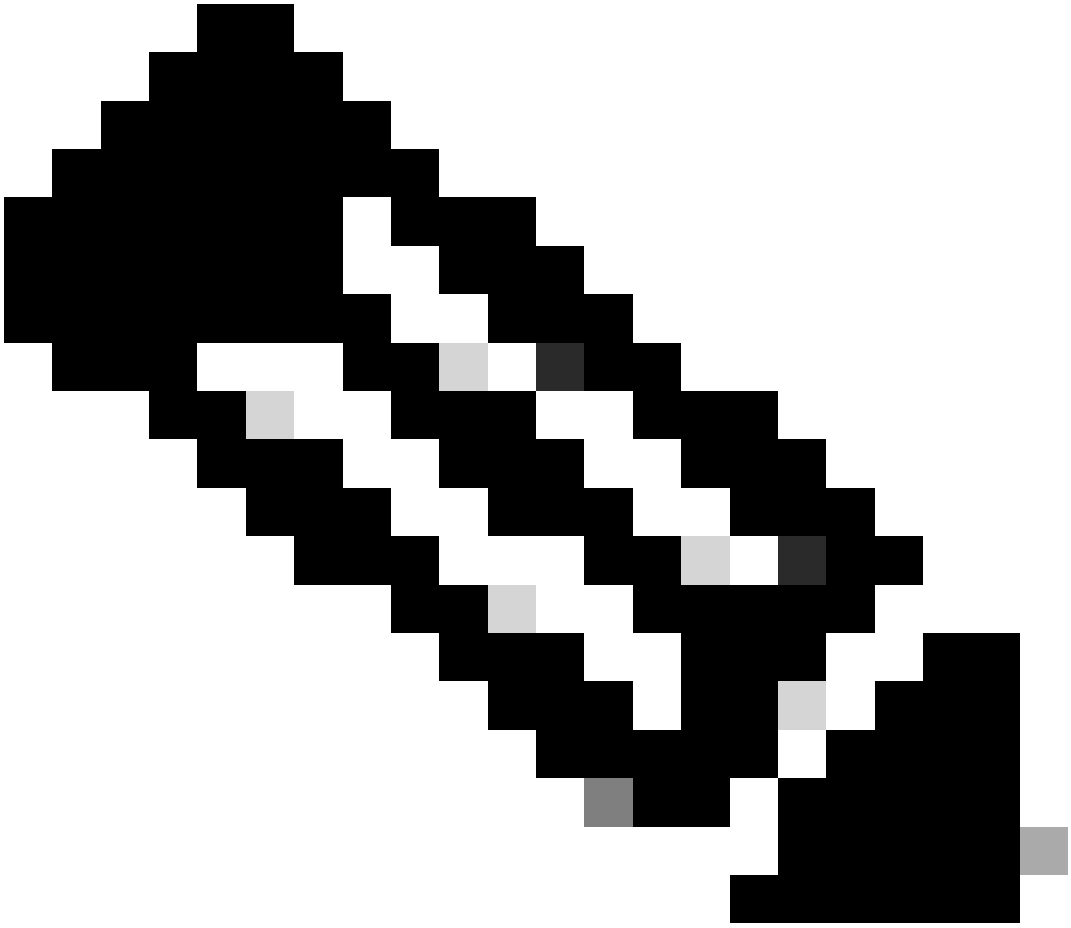
- Carregue o arquivo Lua em Critérios de detecção. Salve e ative o detector.



Reinspect Enabled v/s Disabled (Reinspecionar habilitado v/s desabilitado)



- Os dois eventos mostram o início da conexão v/s e o fim da conexão quando a reinspecção está habilitada.



Observação: o que observar:

1. As "Equipes HTTPS, Webex e Webex" são identificadas pela API no início da conexão. Como a reinspecção é verdadeira, a descoberta de aplicativos continua e as appIds são atualizadas para ‘HTTPS, SSL Client e Gyazo Teams’.

2. Observe o número de pacotes do iniciador e do respondente. Os métodos de detecção de aplicativos regulares exigem muito mais pacotes do que a API.

Solução de problemas/diagnóstico

Visão Geral do Diagnóstico

- Novos registros são adicionados à depuração de identificação de aplicativo de suporte do sistema para indicar se algum aplicativo foi encontrado pela 1ª API de detecção de pacote.
- Os registros também mostram se o usuário escolheu a reinspecção de tráfego.
- O conteúdo do arquivo de detector de lua carregado pelo usuário pode ser encontrado no FTD em /var/sf/appid/custom/lua/<UUID> .
- Quaisquer erros no arquivo lua são despejados no FTD no arquivo /var/log/messages no momento da ativação do detector.

CLI: system support application-identification-debug

<#root>

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

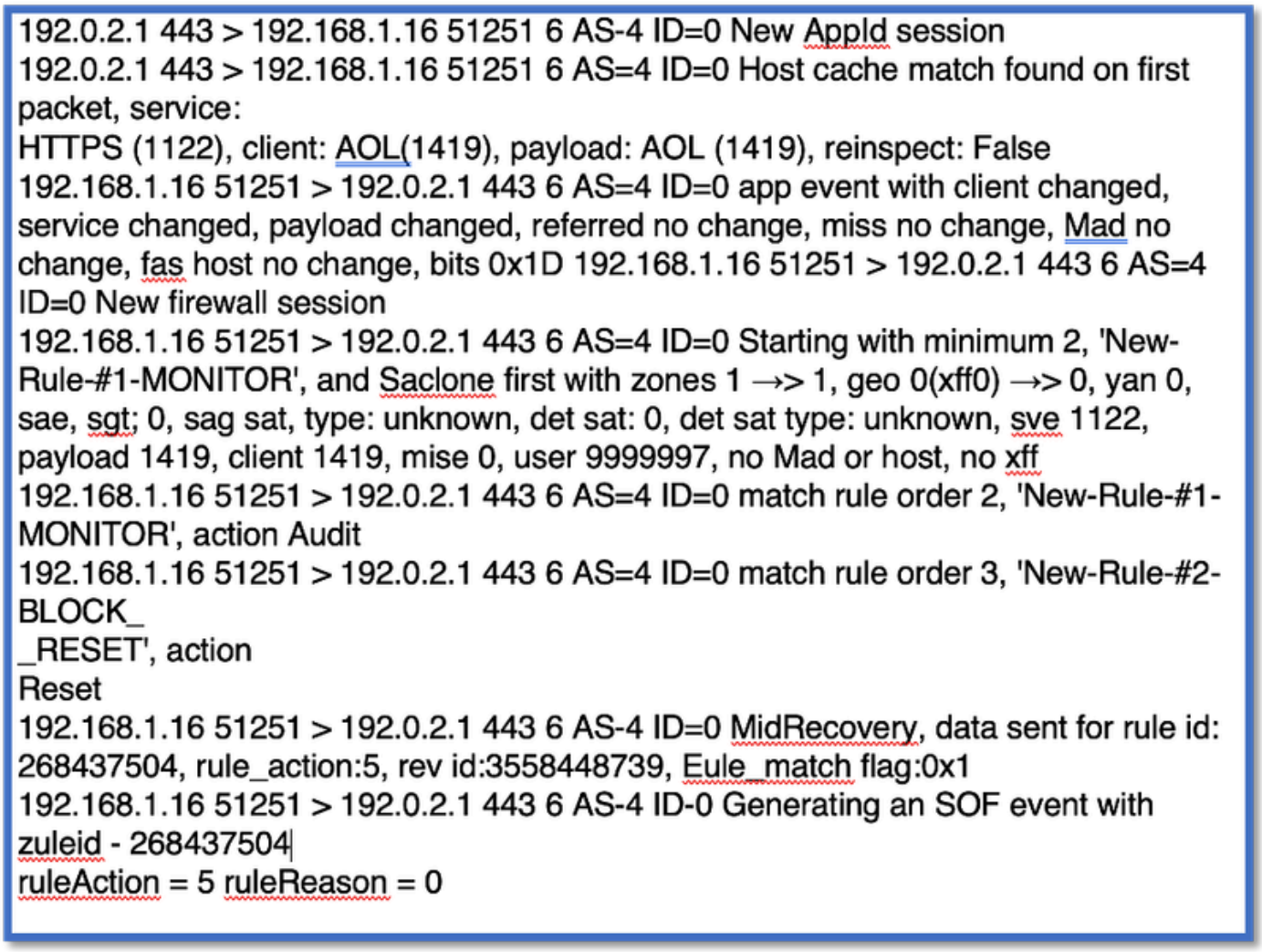
192.0.2.1 443 -> 192.168.1.16 51251 6 AS=4 ID=0 Host cache match found on first packet, service: HTTPS(1

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



Local do Conteúdo de Detectores Lua do AppID

Para confirmar se o Detector Lua com essa nova API existe no Dispositivo/FTD, você pode verificar se a API addHostFirstPktApp está sendo usada nas 2 pastas do detector de aplicativos:

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

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

Por exemplo:grep addHostFirstPktApp * em cada pasta.

Exemplos de problemas:

- Problema: Detector Lua personalizado não ativado no FMC.

Local a ser verificado: /var/sf/appid/custom/lua/

Resultado esperado: um arquivo para cada detector de aplicativo personalizado ativado no FMC deve existir aqui. Verifique se o conteúdo corresponde ao arquivo lua carregado.

- Problema: O arquivo de detector de lua carregado tem erros.

Arquivo a ser verificado: /var/log/messages on FTD

Log de erros:

<#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

Passos de Troubleshooting

Problema: aplicativos não identificados corretamente para o tráfego que vai para a porta e o endereço IP definidos pelo usuário.

Etapas para solucionar problemas:

- Verifique se o detector de lua está definido corretamente e ativado no FTD.
  - Verifique o conteúdo do arquivo lua no FTD e verifique se nenhum erro é visto na ativação.
- Verifique o IP destino, a porta e o protocolo do primeiro pacote na sessão de tráfego.
  - Ele pode corresponder aos valores definidos no detector de lua.
- Verifique o comando system-support-application-identification-debug.

- Procure a linha Host cache match found on first packet. Se ela estiver ausente, isso indica que nenhuma correspondência foi encontrada pela API.

Detalhes das limitações, problemas comuns e soluções alternativas

Na versão 7.4, não há interface do usuário para usar a API. O suporte à interface do usuário seria adicionado em versões futuras.

Histórico das revisões

Revisão	Data de publicação	Comentários
1.0	18-jul-2024	Versão inicial

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

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

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