METODOLOGIAS E ASSINATURAS PARA O SISTEMA DE RASTREAMENTO DE FLUXOS

Testando a segurança de pacotes contendo dados de login. monitoramento de senha no roteador; localizando dados de login sem criptografia; possibilidade da captura de senha de administrador do roteador durante alteração. **Filtros Utilizados:**

http.request.method == "GET" or http.request.method == "POST"

http.request.method == "POST"

urlencoded-form.key

urlencoded-form.key == "USER" or urlencoded-form.key == "usuário"

Ataque Land

envio de pacotes com endereços IP e portas de origem e destino iguais; máquina responde e entra em loop (travamento/falha).

Filtro Utilizado:

(ip.src == ip.dst) && (tcp.srcport == tcp.dstport || udp.srcport == udp.dstport)

Ataques na Camada 7

Camada responsável pela interface direta de inserção/recepção de dados; servidores da Web possui função de pesquisa, diálogo de registro do usuário, função semelhante, que aciona uma resposta demorada no back-end; invasor pode identificar alvos adequados examinando tempo de resposta HTTP, sites podem ser afetados por um número muito pequeno de solicitações HTTP paralelas que acionam pesquisas, processam dados de log, entre outros.

Filtro Utilizado:

http.request.url

SYN Flood

Forma de ataque de negação de serviço (DoS); cliente envia diversos pacotes SYN para o servidor; servidor responde à requisição com um pacote SYN-ACK; cliente não retorna o ACK para confirmar a conexão; as portas do servidor permanecem abertas, sobrecarregando o servidor e congestionando a rede.

Filtro aplicado:

tcp.flags.syn == 1 and tcp.flags.ack == 0

tcp.flags.syn == 1 and tcp.flags.ack == 1

OBSERVAÇÕES:

As indicações de tipo de tráfego, N para 1, 1 para N, N para N e 1 para 1 referem-se a características de 1 hosts estar enviando dados para um ou mais host de destino. A

Scans ou varreduras

Scans são varreduras realizadas por atacantes para obter informações de um ambiente de rede. As assinaturas de varreduras podem detectar tanto varreduras em rede (tentativa de determinar quais são os hosts ativos em uma rede) quanto varreduras em host (tentativa de determinar serviços ativos em um host).

A metodologia desta assinatura consiste em detectar uma grande quantidade de fluxos apenas com a flag SYN habilitada. Isto indica que algum host tentou abrir uma conexão, sendo que no caso de varreduras são tentativas sem sucesso. Como uma conexão gera dois fluxos, um para cada sentido, o conjunto de fluxos do sentido atacante-alvo terá fluxos apenas com a flag SYN. A figura mostra uma assinatura no sistema para detectar varreduras em um host. Dentre as restrições pode-se ver um tráfego 1 para 1, com no mínimo 50 portas de destino diferentes, tempo de duração do fluxo igual a zero, a flag SYN habilitada e considerando todos os protocolos. A restrição de endereço de destino visa analisar as varreduras apenas quando o destino se encontra dentro da rede do Instituto. No entanto, também é possível criar uma assinatura em que esta restrição aparecerá no endereço de origem, ocasião esta que detectará hosts de dentro do Instituto realizando varreduras.

Scan em host

Id: 3
Descrição: Detecta uma varredura em um host, em busca de serviços abertos.
Como detectar (metodologia da assinatura): Procura por uma grande quantidade de fluxos para um único host, em várias portas diferentes, caracterizados pela flag SYN.
Frequência: Comum (várias vezes ao dia).
Criada em: 2009-05-12 10:57:17
Última detecção em: 2009-05-31 21:25:00

Passo: 1 **Código da operação:** 0

Tipo de tráfego: 1 para 1.
Nenhum agrupamento de portas.
Campos selecionados: Srcaddr, Dstaddr.

Contar a quantidade de portas destino distintas. Detectar apenas se a quantidade for >= 50.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: last - first = 0.
Dstaddr wildcard 200.145.%%%%%%%%.
Tcp_flags (fluxos que possuam APENAS as flags): SYN.
Protocolo: Todos.

P2P

P2P é a nomenclatura utilizada para denominar aplicações de compartilhamento de arquivos tais como Emule, aMule, dentre outras. Estas aplicações se conectam em redes distribuídas como Gnutella, eDonkey e FastTrack. Estas aplicações geram problemas tais como o uso indevido de banda além de violações de políticas de rede e direitos autorais em arquivos protegidos.

Para a detecção destas aplicações foram geradas duas assinaturas devido ao comportamento distinto em dois casos diferentes. O primeiro caso é quando a aplicação executa em um host sem filtragem de portas. O segundo é quando este host está protegido por um sistema de firewall (com tradução de endereços, processo conhecido como NAT). Analisando as amostras coletadas, é possível perceber a diferença entre os casos e gerar as duas assinaturas.

Nos dois casos, o tráfego gerado é do tipo 1 para N a partir do host que executa o cliente P2P. Normalmente este tráfego envolve mais de 150 destinos. Os protocolos utilizados são o UDP e o TCP, sendo o primeiro para troca de informações sobre a rede distribuída e o segundo para a transferência dos arquivos. A figura mostra a assinatura para a detecção de P2P sem filtragem de portas.

Para o caso com filtragem de portas, um passo é inserido para detectar uma repetição de fluxos com uma quantidade de pacotes e bytes idêntica. Estes fluxos referem-se a pacotes filtrados pelo Firewall, ocorrendo em apenas um sentido.

P2P

- As transferências utilizam o protocolo TCP, com portas origem e destinos maiores que 1023 sendo que a média de pacotes foi sempre maior que 10, enquanto a média de bytes por fluxos maior que 5000.
- O segundo passo consiste das tarefas de conexão, atualização e buscas nas redes distribuídas, todas utilizando TCP. Para estas tarefas, além do tráfego 1 para N foi verificado que as aplicações utilizam uma mesma porta de origem. Assim, para cada host retornado como suspeito no primeiro passo é procurado o segundo, cujas restrições são portas origem e destino maiores que 1023, quantidade de endereços e portas destino maiores que 150.

Filtro Utilizado:

sudo tcpdump greater 4999 && tcp && dst src port > 1023 (p2p)

P2P

Id: 5
Descrição: Atividade de compartilhamento de arquivos com redes do tipo Kazaa, Emule, FastTrack, entre outras, também conhecidas como aplicações P2P
Como detectar (metodologia da assinatura): Uma aplicação P2P gera dois padrões de fluxos. O primeiro consiste do startup e mensagens de gerenciamento, todas UDP. O segundo são conexões TCP que representam a transferência de arquivos entre os hosts da rede distribuída. Primeiramente verifica-se se houve o padrão de startup e em seguida detecta-se as características da transferência de arquivos
Frequência: Comum (várias vezes ao dia).
Criada em: 2009-04-14 11:33:47
Última detecção em: 2009-05-31 07:25:00

Passo: 1 **Código da operação:** 0

Tipo de tráfego 1 para N.
Característica de serviço: Mesma porta de origem.
Campos selecionados: Srcaddr, Srcport.

Contar a quantidade de endereços destino distintos. Detectar apenas se a quantidade for >= 150.
Contar a quantidade de portas destino distintas. Detectar apenas se a quantidade for >= 150.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: last - first <= 6.
Srcaddr wildcard 200.145.%%%%%%%.
Protocolo: UDP.

Passo: 2 **Código da operação:** 0

Tipo de tráfego 1 para N.
Característica de serviço: Mesma porta de origem.
Campos selecionados: Srcaddr, Srcport.

Contar a quantidade total de fluxos. Detectar apenas se a quantidade for >= 5.
Contar a quantidade de endereços destino distintos. Detectar apenas se a quantidade for >= 5.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: last - first >= 50.
Srcaddr = srcaddr do passo 1.
Srcport = srcport do passo 1.
Protocolo: TCP.

Bittorrent

Este protocolo é utilizado atualmente para troca de arquivos entre hosts distribuídos, mas com operações iniciais centralizadas em sistemas conhecidos como trackers. Um cliente normalmente estabelece uma conexão inicial com o tracker e obtém informações sobre outros hosts que possuem o mesmo arquivo. A partir de então, o cliente passa a gerar tráfego 1 para N a fim de obter partes do arquivo.

A geração da assinatura de Bittorrent foi a mais complexa dentre todas geradas, devido a quantidade de detalhes necessários para descrever este evento. Ela é composta de 7 passos que envolvem as seguintes características:

Tráfego 1 para N, normalmente com mais de 20 hosts destino;

O cliente utiliza várias portas origem diferentes para a troca de arquivos;

As conexões possuem várias portas destino diferentes;

Repetição no número de bytes e pacotes nos fluxos enviados pelo cliente;

O cliente repete a utilização de alguma porta de origem (esta porta é configurada no cliente e utilizada para que outros hosts entrem em contato para troca de controles).

Para cada uma destas características foram definidos limiares considerados mínimos para que a aplicação seja detectada. A figura mostra a assinatura de Bittorrent com os limiares utilizados em cada um dos passos.

Monitorando o uso de BitTorrent na rede.

identifica download torrent;

Filtro Utilizado:

udp contains BitTorrent or tcp contains BitTorrent or bittorrent

Bittorrent

Id: 2
Descrição: Atividade de compartilhamento de arquivos baseada no padrão Torrent
Como detectar (metodologia da assinatura): Detecta-se várias características do padrão como tráfego 1-N, repetições de portas origem e destino e repetições da quantidade de pacotes e bytes em cada fluxo
Frequência: Comum (várias vezes ao dia).
Criada em: 2009-04-02 11:44:33
Última detecção em: 2009-07-14 15:50:00

Passo: 1 **Código da operação:** 0

Tipo de tráfego 1 para N.
Nenhum agrupamento de portas.
Campos selecionados: Srcaddr.

Contar a quantidade de endereços destino distintos. Detectar apenas se a quantidade for >= 20.
Contar a quantidade de portas origem distintas. Detectar apenas se a quantidade for >= 45.
Contar a quantidade de portas destino distintas. Detectar apenas se a quantidade for >= 25.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: Srcaddr wildcard 200.145.%%%%%%%%.
Protocolo: TCP.

Passo: 2 **Código da operação:** 0

Nenhum agrupamento de tráfego.
Nenhum agrupamento de portas.
Campos selecionados: Srcaddr.

Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: Srcaddr = srcaddr do passo 1.
Bytes / Pacotes repetindo >= 25.
Protocolo: TCP.

Passo: 3 **Código da operação:** 0

Tipo de tráfego 1 para N.
Característica de serviço: Mesma porta de origem.
Campos selecionados: Srcaddr, Srcport.

Contar a quantidade total de fluxos. Detectar apenas se a quantidade for >= 4.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: Srcaddr = srcaddr do passo 2.
Protocolo: TCP.

Passo: 4 **Código da operação:** 0

Tipo de tráfego 1 para N.
Característica de serviço: Mesma porta de destino.
Campos selecionados: Srcaddr, Dstport.

Contar a quantidade total de fluxos. Detectar apenas se a quantidade for >= 4.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: Srcaddr = srcaddr do passo 3.
Protocolo: TCP.

Passo: 5 **Código da operação:** 0

Tipo de tráfego: N para 1.
Nenhum agrupamento de portas.
Campos selecionados: Dstaddr.

Contar a quantidade de endereços origem distintos. Detectar apenas se a quantidade for >= 20.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: Dstaddr = srcaddr do passo 4.
Protocolo: TCP.

Passo: 6 **Código da operação:** 0

Tipo de tráfego: N para 1.
Característica de serviço: Mesma porta de destino.
Campos selecionados: Dstaddr, Dstport.

Contar a quantidade total de fluxos. Detectar apenas se a quantidade for >= 10.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: Dstaddr = dstaddr do passo 5.
Protocolo: TCP.

Passo: 7 **Código da operação:** 0

Nenhum agrupamento de tráfego.
Nenhum agrupamento de portas.
Campos selecionados: Dstaddr.

Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: last - first = 0.
Dstaddr = dstaddr do passo 6.
Bytes / Pacotes repetindo >= 4.
Protocolo: TCP.

Ataques no serviço de SSH

Detecta ataques de força bruta ou dicionário, em que vários usuários e senha são testados continuamente em hosts que possuem o serviço de SSH (Secure Shell protocol). Uma vez que usuários e senhas são normalmente cadeia de caracteres com tamanhos parecidos, os fluxos gerados em cada tentativa apresentam certo padrão de comportamento quanto aos campos que tratam da quantidade de bytes e pacotes. A assinatura considera um mínimo de 15 tentativas, com duração de cada fluxo entre 5 e 12 segundos, média de pacotes entre 7 e 17, a quantidade de bytes por pacote entre 90 e 100 bytes, protocolo TCP e a flag SYN habilitada. A figura mostra a assinatura de detecção de ataques no SSH.

Embora na descrição da metodologia da assinatura seja mencionada a restrição da porta 22 como destino, cabe mencionar que os testes foram realizados sem esta restrição. Desta forma, fica evidente a característica do modelo de detectar eventos pelo comportamento causado nos fluxos do ambiente. Não é necessário que o serviço seja executado na porta 22, padrão do SSH. Qualquer que seja a porta utilizada, se o ataque de dicionário ocorrer ele será detectado pelas características dos fluxos gerados. Esta capacidade tem sido alvo de diversas pesquisas dentro da classificação de tráfego, tal como o modelo discutido em (Karagiannis; Papagiannaki; Faloutsos, 2005).

SSH Brute Force

Id: 1
Descrição: Ataque de força bruta contra o serviço SSH na porta 22. O ataque consiste em várias tentativas de autenticar um usuário e ganhar acesso ao sistema. São testados vários usuários e senhas
Como detectar (metodologia da assinatura): Procura-se por uma grande quantidade de fluxos que possuam a porta destino igual a 22, tenham uma duração entre 1 e 55 segundos, protocolo TCP, com uma média bytes/fluxo entre 50 e 1500, uma média de pacotes/fluxo entre 1 e 16 e a flag SYN habilitada. Se 15 tentativas destas forem encontradas nos últimos 5 minutos o evento é detectado
Frequência: Comum (várias vezes ao dia).
Criada em: 2009-03-03 16:54:32
Última detecção em: 2009-07-14 15:50:00

Passo: 1 **Código da operação:** 0
Tipo de tráfego: 1 para 1.
Característica de serviço: Mesma porta de destino.
Campos selecionados: Srcaddr, Dstaddr, Dstport.

Contar a quantidade total de fluxos. Detectar apenas se a quantidade for >= 15.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: Média de last - first entre 3 e 90.
Dstaddr wildcard 200.145.%%%%%%%%.
Média de pacotes entre 7 e 17.
Bytes / Pacotes entre 90 e 115.
Tcp_flags (fluxos com no MÍNIMO as flags): SYN.
Protocolo: TCP.

Skype

Esta assinatura detecta usuários utilizando o softphone Skype quando o usuário conecta e realiza alguma chamada de voz. Testes foram realizados utilizando tanto o Skype para sistemas Linux quanto para sistemas Windows, sendo ambos detectados. A assinatura é dividida em dois passos, sendo um correspondente à inicialização da aplicação (comunicação com servidores de autenticação) e outro à chamada de voz. O primeiro passo da assinatura é a detecção da chamada de voz caracterizada pelo uso do protocolo UDP, tráfego 1 para 1, tempo de duração do fluxo maior que 30 segundos, com uma média entre 130 e 320 bytes por pacote. As portas utilizadas são normalmente maiores que 1023. Detectado um padrão como este (possível chamada de voz em andamento), o segundo passo procura pela inicialização da aplicação, caracterizada por um tráfego 1 para N (vários servidores Skype), fluxos utilizando sempre uma mesma porta de origem e com duração menor que 60 segundos. A **Error! Reference source not found.** ilustra a assinatura para detecção do Skype. É possível observar pela figura que o passo 1 descreve a chamada de voz e o passo 2 o processo de inicialização e autenticação, ocorrendo uma inversão na ordem cronológica destes eventos. Isto ocorre para se otimizar o tempo de análise da assinatura. Uma vez que para cada resultado do passo 1 deve-se testar a ocorrência do passo 2, quanto menos resultados intermediários o passo 1 gerar mais rápido será o teste da assinatura, pois será menor o número de análises para o passo 2. Em casos em que os passos possibilitam esta inversão, como é o do Skype, esta pode ser realizada, otimizando assim a análise online dos fluxos.

Skype

Id: 4
Descrição: Detecção da aplicação Skype que tenha feito autenticação e tenha efetuado alguma chamada de voz
Como detectar (metodologia da assinatura): O primeiro passo consiste em detectar possíveis chamadas de voz em que a característica do tráfego é possuir portas altas e determinado padrão de bytes e pacotes. Para cada um destes possíveis hosts realizando uma chamada são procurados os fluxos correspondentes ao startup desta aplicação, em que ocorre uma repetição de portas origem e um padrão de bytes e pacotes
Frequência: Normal (poucas vezes ao dia).
Criada em: 2009-04-05 13:43:37
Última detecção em: 2009-05-29 21:30:00

Passo: 1 **Código da operação:** 0

Tipo de tráfego: 1 para 1.
Característica de serviço: Mesma porta de origem.
Campos selecionados: Srcaddr, Dstaddr, Srcport, Dstport.

Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 5 minutos.
Restrição de tempo: last - first >= 30.
Srcaddr wildcard 200.145.%%%%%%%%.
Srcport > 1023.
Dstport > 1023.
Bytes / Pacotes entre 130 e 320.
Protocolo: UDP.

Passo: 2 **Código da operação:** 0

Tipo de tráfego 1 para N.
Característica de serviço: Mesma porta de origem.
Campos selecionados: Srcaddr, Srcport.

Contar a quantidade de endereços destino distintos. Detectar apenas se a quantidade estiver entre 50 e 500.
Contar a quantidade de portas destino distintas. Detectar apenas se a quantidade estiver entre 50 e 500.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 30 minutos (padrao).
Restrição de tempo: last - first <= 60.
Srcaddr = srcaddr do passo 1.
Srcport = srcport do passo 1.
Protocolo: UDP.

Worms

O MyDoom é um artefato malicioso destinado aos sistemas Windows, capaz de se propagar automaticamente por uma rede. Uma das primeiras versões surgiu em 2004 e tornou-se popular pela rapidez na disseminação. Após a análise dos fluxos deste artefato, foi gerada uma assinatura capaz de detectar um host sendo infectado, passando a propagar o artefato. Existe uma diferença básica na metodologia de geração da assinatura para artefatos como este. Uma vez que se trata de softwares destrutivos, para cada amostra coletada todo o sistema operacional era reinstalado. Após a reinstalação, o sistema era infectado e os fluxos gerados pela atividade do artefato exportados e coletados.

A assinatura é composta de 4 passos: 1) fluxos que possuam exatamente 7 pacotes, com 2034 bytes cada, com a porta destino 135, protocolo TCP, as flags Syn, Fin, Psh e Ack habilitadas, possuindo um host dentro do ambiente monitorado como destino; 2) utilizando cada endereço destino do passo 1 como origem, detectar fluxos com 5 pacotes, 268 bytes, a mesma combinação de flags e protocolo TCP; 3) os dois passos anteriores representam uma infecção bem sucedida e o terceiro passo consiste em detectar consultas a um servidor de nomes realizadas pelo host infectado, ou seja, uma repetição de no mínimo 30 fluxos, com porta destino 53 e protocolo UDP; 4) o passo final consiste na busca do host tentando propagar o artefato, sendo detectada a repetição de fluxos com porta destino 135, flag Syn habilitada, protocolo TCP, em que a média de bytes por pacote seja 48 (dOctets/dPkts) em um tráfego 1 para N, característico de uma varredura. A figura mostra esta estrutura.

A complexidade na geração de assinaturas para artefatos maliciosos decai sob a dificuldade em se analisar o comportamento real destes softwares maliciosos. A geração das amostras de fluxos não é trivial como em outros eventos devido a capacidade destrutiva destes artefatos. Outro fator que aumenta a complexidade é a necessidade de se reinstalar os sistemas operacionais a cada amostra. Uma vez infectado o sistema deve ser descartado para garantir que novas amostras contenham os mesmos comportamentos, desde a infecção do host. Desta forma a assinatura descreverá tanto a etapa de infecção quanto a do comportamento do host já infectado, e não apenas esta

última, como seria observado caso o sistema operacional não fosse reinstalado e uma nova amostra fosse gerada.

Worms Mydoom e variantes

Conhecido como W32.MyDoom@mm, Novarg, Mimail.R e Shimgapi, vírus que afeta Windows; propaga automaticamente por uma rede; assinatura capaz de detectar host infectado.

- fluxos que possuam exatamente 7 pacotes, com 2034 bytes cada, com a porta destino 135, protocolo TCP, as flags Syn, Fin, Psh e Ack habilitadas, tendo como destino um host dentro do ambiente monitorado;
- utilizando cada endereço destino do passo 1 como origem, detectar fluxos com 5 pacotes, 268 bytes, a mesma combinação de flags e protocolo TCP;
- os dois passos anteriores representam uma infecção bem sucedida e o terceiro passo consiste em detectar consultas a um servidor de nomes realizadas pelo host infectado, ou seja, uma repetição de no mínimo 30 fluxos, com porta destino 53 e protocolo UDP;
- o passo final consiste na busca do host tentando propagar o artefato, sendo detectada a repetição de fluxos com porta destino 135, flag Syn habilitada, protocolo TCP.

Filtro Utilizado:

```
sudo tcpdump greater 2033 && tcp && dst port 135 (mydoom)
```

MyDoom

Id: 3
Descrição: Propagação do Worm MyDoom, variante A.
Como detectar (metodologia da assinatura): Consiste na detecção de 4 passos, desde a infecção dos host, transmissão de uma sequencia de bytes e pacotes, consultas de DNS e tentantiva de disseminação automática.
Frequência: Raro (não ocorre diariamente).
Criada em: 2009-07-28 22:26:30
Última detecção em: 2009-01-01 00:00:00

Passo: 1 **Código da operação:** 0

Nenhum agrupamento de tráfego.
Nenhum agrupamento de portas.
Campos selecionados: Dstaddr, Srcport, Dstport.

Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 10 minutos.
Restrição de tempo: Dstaddr wildcard 200.145.%.
Dstport = 135.
Tcp_flags (fluxos com no MÍNIMO as flags): ACK, PSH, SYN, FIN.
Protocolo: TCP.

Passo: 2 **Código da operação:** 0

Nenhum agrupamento de tráfego.
Nenhum agrupamento de portas.
Campos selecionados: Srcaddr.

Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 10 minutos.
Restrição de tempo: Srcaddr = dstaddr do passo 1.
Tcp_flags (fluxos com no MÍNIMO as flags): ACK, PSH, SYN, FIN.
Protocolo: TCP.

Passo: 3 **Código da operação:** 0

Nenhum agrupamento de tráfego.
Nenhum agrupamento de portas.
Campos selecionados: Srcaddr, Dstport.

Contar a quantidade total de fluxos. Detectar apenas se a quantidade for >= 20.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 10 minutos.
Restrição de tempo: Srcaddr = srcaddr do passo 2.
Dstport = 53.
Protocolo: UDP.

Passo: 4 **Código da operação:** 0

Tipo de tráfego 1 para N.
Nenhum agrupamento de portas.
Campos selecionados: Srcaddr, Dstport.

Contar a quantidade de endereços destino distintos. Detectar apenas se a quantidade for >= 10.
Executar a cada: 5 minutos.
Intevalo de tempo: Ultimos 10 minutos.
Restrição de tempo: Srcaddr = srcaddr do passo 3.
Dstport = 135.
Bytes / Pacotes = 48.
Tcp_flags (fluxos com no MÍNIMO as flags): SYN.
Protocolo: TCP.