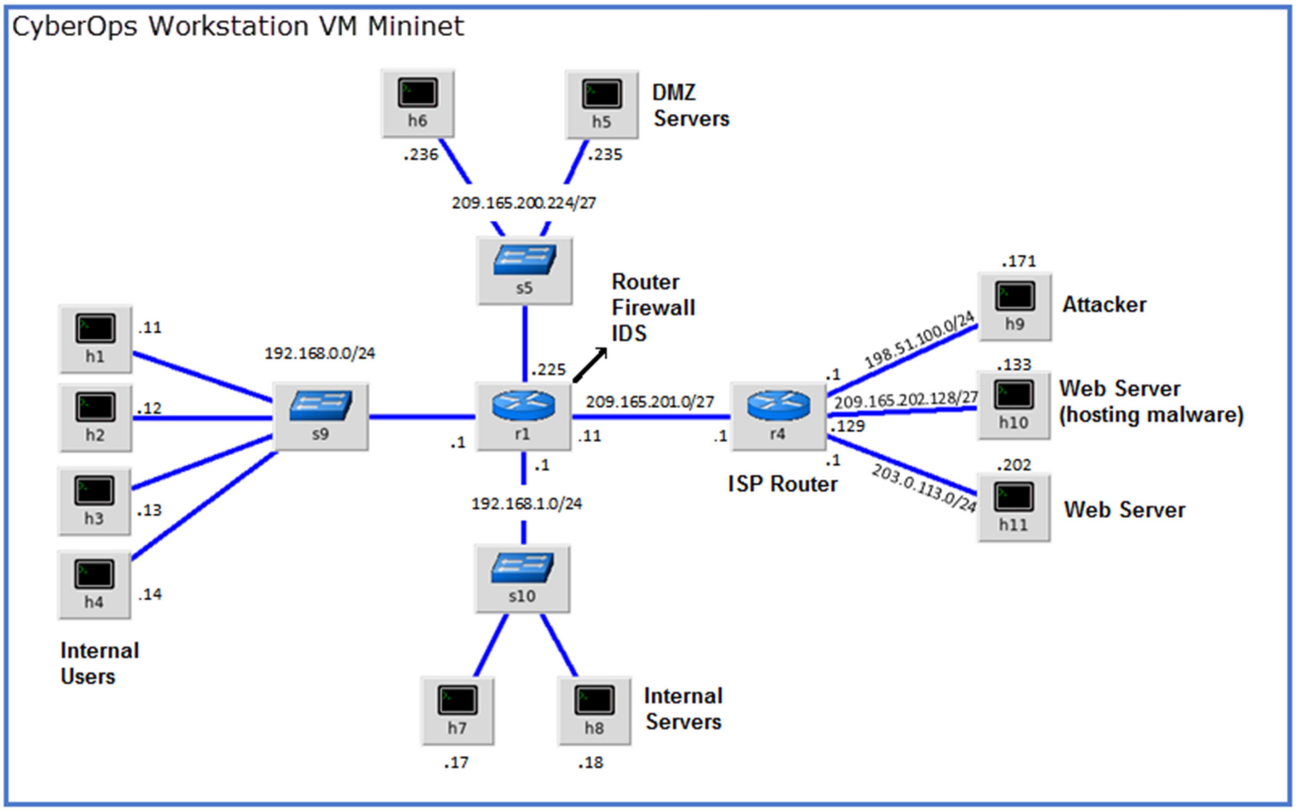
Laboratório - Regras de Snort e Firewall

Topologia



Objetivos

Parte 1: Preparando o Ambiente Virtual

Parte 2: Firewall e logs IDS

Parte 3: Encerrar e Limpar o Processo Mininet

Histórico/Cenário

Em uma rede de produção segura, os alertas de rede são gerados por vários tipos de dispositivos, como dispositivos de segurança, firewalls, dispositivos IPS, roteadores, switches, servidores e muito mais. O problema é que nem todos os alertas são criados igualmente. Por exemplo, alertas gerados por um servidor e alertas gerados por um firewall serão diferentes e variam em conteúdo e formato.

Neste laboratório, para se familiarizar com regras de firewall e assinaturas IDS.

Recursos necessários

- Máquina Virtual CyberOps Workstation
- Conexão com a Internet

Observação: Neste laboratório, a VM CyberOps Workstation é um contêiner para armazenar o ambiente Mininet mostrado na Topologia. Se um erro de memória for recebido em uma tentativa de executar qualquer comando, saia da etapa, vá para as configurações da VM e aumente a memória. O padrão é 1 GB; tente 2 GB.

Instruções

Parte 1: Preparando o Ambiente Virtual

a. Inicie o **Oracle VirtualBox** etransmita o **CyberOps Workstation** para o modo Bridged, se necessário. Selecione **Machine > Settings > Network**. Em **Attached To**, selecione **Bridged Adapter** (ou, se estiver usando WiFi com um proxy, talvez seja necessário um **NAT adapter**) e clique em **OK**.

b. Inicie a **VM CyberOps Workstation**, abra um terminal e configure sua rede executando o script **configure_as_dhcp.sh** .

Como o script requer privilégios de superusuário, forneça a senha para o usuário **analyst**

```
[analyst @secOps ~] $ sudo./lab.support.files/scripts/configure_as_dhcp.sh
[sudo] password for analyst:
[analyst @secOps ~] $
```

c. Use o comando **ifconfig** para verificar se a **VM CyberOps Workstation** agora tem um endereço IP em sua rede local. Você também pode testar a conectividade com um servidor público da Web fazendo ping em www.cisco.com. Use **Ctrl+C** para parar os pings.

```
[analyst @secOps ~] $ ping www.cisco.com
PING e2867.dsca.akamaiedge.net (23.204.15.199) 56(84) bytes of data.
64 bytes from a23-204-15-199.deploy.static.akamaitechnologies.com
(23.204.15.199): icmp_seq=1 ttl=54 time=28.4 ms
64 bytes from a23-204-15-199.deploy.static.akamaitechnologies.com
(23.204.15.199): icmp_seq=2 ttl=54 time=35.5 ms
^C
--- e2867.dsca.akamaiedge.net ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/máx/mdev = 28.446/32.020/35,595/3.578 ms
```

Parte 2: Logs de firewall e IDS

Firewalls e sistemas de detecção de intrusões (IDS) são frequentemente implantados para automatizar parcialmente a tarefa de monitoramento de tráfego. Ambos os firewalls e IDSs correspondem ao tráfego de entrada em relação às regras administrativas. Os firewalls costumam comparar o cabeçalho do pacote com um conjunto de regras, enquanto os IDSs costumam usar a carga do pacote para comparação do conjunto de regras. Como firewalls e IDSs aplicam as regras predefinidas a diferentes partes do pacote IP, IDS e regras de firewall têm estruturas diferentes.

Embora haja uma diferença na estrutura de regras, algumas semelhanças entre os componentes das regras permanecem. Por exemplo, as regras de firewall e IDS contêm componentes correspondentes e componentes de ação. As ações são tomadas após uma correspondência ser encontrada.

- **Componente de correspondência** - especifica os elementos de pacote de interesse, como: origem do pacote; destino do pacote; portas e protocolos da camada de transporte; e dados incluídos na carga útil do pacote.
- **Componente de ação** - especifica o que deve ser feito com esse pacote que corresponde a um componente, como: aceitar e encaminhar o pacote; descartar o pacote; ou enviar o pacote para um conjunto de regras secundário para inspeção adicional.

Um design de firewall comum é descartar pacotes por padrão enquanto especifica manualmente qual tráfego deve ser permitido. Conhecido como dropping-by-default, este design tem a vantagem de proteger a rede contra protocolos e ataques desconhecidos. Como parte desse design, é comum registrar os eventos de pacotes descartados, uma vez que estes são pacotes que não foram explicitamente permitidos e, portanto, infringem as políticas da organização. Tais eventos devem ser registados para análise futura.

Etapa 1: Monitoramento de registro IDS em tempo real

a. Na **VM CyberOps Workstation**, execute o script para iniciar o **mininet**.

```
[analyst @secOps ~] $
sudo./lab.support.files/scripts/cyberops_extended_topo_no_fw.py
[sudo] password for analyst:
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
R1 R4 H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11
*** Starting controllers
*** Starting switches
*** Add routes
*** Post configure switches and hosts
*** Starting CLI:
mininete>
```

O prompt do **mininet** deve ser exibido, indicando que o **mininet** está pronto para comandos.

b. No prompt do **mininet**, abra um shell no **R1** usando o comando abaixo:

```
mininet xterm R1
mininet>
```

O shell **R1** abre em uma janela de terminal com texto preto e fundo branco. Qual usuário está logado nesse shell? Qual é o indicador disso?

c. No shell do **R1**, inicie o IDS baseado em Linux, Snort.

```
[root @secOps analyst] #./lab.support.files/scripts/start_snort.sh
Executando no modo IDS
—== Inicializando Snort ==—
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
<output omitted>
```

Observação: Você não verá um prompt, pois o Snort está sendo executado nesta janela. Se, por algum motivo, o Snort parar de funcionar e o **prompt** [root @secOps analist] # for exibido, execute novamente o script para iniciar o Snort. Snort deve estar executando para capturar alertas mais tarde no laboratório.

d. No prompt do **mininet da VM CyberOps Workstation**, abra shells para hosts **H5** e **H10**.

```
mininet xterm H5
mininet xterm H10
mininet>
```

e. **O H10** simulará um servidor na Internet que hospeda malware. Em **H10**, execute o script mal_server_start.sh para iniciar o servidor.

```
[root @secOps analyst] #./lab.support.files/scripts/mal_server_start.sh
[root@secOps analyst]#
```

f. No **H10**, use **netstat** com as opções **-tunpa** para verificar se o servidor web está sendo executado. Quando usado como mostrado abaixo, **netstat** lista todas as portas atualmente atribuídas aos serviços:

```
[root @secOps analyst] # netstat -tunpa
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp 0 0 0.0.0.0:6666 0.0.0.0:* LISTEN 1839/nginx: master
[root@secOps analyst]#
```

Como visto pela saída acima, o servidor web leve **nginx** está sendo executado e ouvindo conexões na porta TCP 6666.

g. Na janela do terminal **R1**, uma instância do Snort está em execução. Para inserir mais comandos no **R1**, abra outro terminal **R1** inserindo o **xterm R1** novamente na janela do terminal da **VM CyberOps Workstation** . Você também pode querer organizar as janelas do terminal para que você possa ver e interagir com cada dispositivo.

h. Na nova guia do terminal **R1**, execute o comando **tail** com a opção **-f** para monitorar o arquivo **/var/log/snort/alert** em tempo real. Este arquivo é onde o snort está configurado para registrar alertas.

```
[root @secOps analyst] # tail -f /var/log/snort/alert
```

Como nenhum alerta ainda foi registrado, o log deve estar vazio. No entanto, se você tiver executado este laboratório antes, as entradas de alerta antigas podem ser mostradas. Em ambos os casos, você não receberá um prompt depois de digitar este comando. Esta janela exibirá alertas à medida que eles acontecem.

i. Em **H5**, use o comando **wget** para baixar um arquivo chamado **W32.Nimda.amm.exe**. Projetado para baixar conteúdo via HTTP, **wget** é uma ótima ferramenta para baixar arquivos de servidores web diretamente da linha de comando.

```
[root @secOps analyst] # wget 209.165.202. 133:6666 /W32.Nimda.amm.exe
--2017-04-28 17:00:04-- http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... connected.
HTTP request sent, awaiting response... 200 OK
Length: 345088 (337K) [application/octet-stream]
Saving to: 'W32.Nimda.Amm.exe'

W32.Nimda.Amm.exe 100%[=========================================>] 337.00K --.-KB/s
in 0.02s

2017-04-28 17:00:04 (16.4 MB/s) - 'W32.Nimda.Amm.exe' saved [345088/345088]

[root@secOps analyst]#
```

Qual porta é usada ao se comunicar com o servidor web de malware? Qual é o indicador?

O arquivo foi completamente baixado?

O IDS gerou algum alerta relacionado ao download do arquivo?

j. Como o arquivo malicioso estava transitando **R1**, o IDS, Snort, foi capaz de inspecionar sua carga útil. A carga correspondeu a pelo menos uma das assinaturas configuradas no Snort e disparou um alerta na segunda janela do terminal **R1** (a guia onde o **tail -f** está sendo executado). A entrada de alerta é mostrada abaixo. Seu carimbo de data/hora será diferente:

```
04/28-17:00:04.092153 [**] [1:1000003:0] Malicious Server Hit! [**] [Priority: 0]
{TCP} 209.165.200.235:34484 -> 209.165.202.133:6666
```

Com base no alerta mostrado acima, quais foram os endereços IPv4 de origem e destino usados na transação?

Com base no alerta mostrado acima, quais foram as portas de origem e destino usadas na transação?

Com base no alerta mostrado acima, quando o download ocorreu?

Com base no alerta mostrado acima, qual foi a mensagem registrada pela assinatura IDS?

Em **H5**, use o comando **tcpdump** para capturar o evento e baixar o arquivo de malware novamente para que você possa capturar a transação. Execute o seguinte comando abaixo iniciar a captura de pacotes:

```
[root @secOps analyst] # tcpdump –i H5-eth0 –w nimda.download.pcap &
[1] 5633
[root@secOps analyst]# tcpdump: listening on H5-eth0, link-type EN10MB (Ethernet),
capture size 262144 bytes
```

O comando acima instrui o tcpdump para capturar pacotes na interface **H5-eth0** e salvar a captura em um arquivo chamado **nimda.download.pcap**.

O **&** símbolo no final diz ao shell para executar o **tcpdump** em segundo plano. Sem este símbolo, o **tcpdump** tornaria o terminal inutilizável enquanto ele estava em execução. Observe o **[1] 5633**; ele indica que um processo foi enviado para segundo plano e seu ID de processo (PID) é 5366. Seu PID provavelmente será diferente.

k. Pressione **ENTER** algumas vezes para recuperar o controle do shell enquanto o **tcpdump** é executado em segundo plano.

l. Agora que o **tcpdump** está capturando pacotes, baixe o malware novamente. Em **H5**, execute novamente o comando ou use a seta para cima para recuperá-lo do recurso de histórico de comandos.

```
[root @secOps analyst] # wget 209.165.202. 133:6666 /W32.Nimda.amm.exe
--2017-05-02 10:26:50-- http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... connected.
HTTP request sent, awaiting response... 200 OK
Length: 345088 (337K) [application/octet-stream]
Saving to: 'W32.Nimda.Amm.exe'

W32.Nimda.amm.exe 100% [====================>] 337.00K -.-KB/s em 0.003s

2017-05-02 10:26:50 (105 MB/s) - 'W32.Nimda.Amm.exe' saved [345088/345088]
```

m. Pare a captura trazendo **tcpdump** para primeiro plano com o comando **fg**. Como **tcpdump** foi o único processo enviado para segundo plano, não há necessidade de especificar o PID. Pare o processo **tcpdump** com **Ctrl+C**. O processo **tcpdump** para e exibe um resumo da captura. O número de pacotes pode ser diferente para sua captura.

```
[root @secOps analyst] # fg
tcpdump -i h5-eth0 -w nimda.download.pcap
^C316 packets captured
316 packets received by filter
0 packets dropped by kernel
[root@secOps analyst]#
```

n. Em **H5**, use o comando **ls** para verificar se o arquivo pcap foi, de fato, salvo no disco e tem tamanho maior que zero:

```
[root @secOps analyst] # ls -l
total 1400
drwxr-xr-x 2 analyst analyst 4096 Sep 26 2014 Desktop
drwx------ 3 analyst analyst 4096 Jul 14 11:28 Downloads
drwxr-xr-x 8 analyst analyst 4096 Jul 25 16:27 lab.support.files
-rw-r--r-- 1 root root 371784 Aug 17 14:48 nimda.download.pcap
drwxr-xr-x 2 analyst analyst 4096 Mar 3 15:56 second_drive
-rw-r--r-- 1 root root 345088 Apr 14 15:17 W32.Nimda.Amm.exe
-rw-r--r-- 1 root root 345088 Apr 14 15:17 W32.Nimda.Amm.exe.1
 [root@secOps analyst]#
```

Observação: Sua lista de diretórios pode ter uma combinação diferente de arquivos, mas você ainda deve ver o arquivo nimda.download.pcap.

Como esse arquivo PCAP pode ser útil para o analista de segurança?

Nota: A análise do arquivo PCAP será realizada em outro laboratório.

Etapa 2: Ajustar regras de firewall com base em alertas IDS

Na Etapa 1, você iniciou um servidor mal-intencionado baseado na Internet. Para impedir que outros usuários acessem esse servidor, é recomendável bloqueá-lo no firewall de borda.

Na topologia deste laboratório, o **R1** não está apenas executando um IDS, mas também um firewall baseado em Linux muito popular chamado **iptables**. Nesta etapa, você bloqueará o tráfego para o servidor mal-intencionado identificado na Etapa 1 editando as regras de firewall atualmente presentes no **R1**.

Nota: Embora um estudo abrangente de **iptables** esteja além do escopo deste curso, **a** lógica básica iptables e a estrutura de regras são bastante simples.

O firewall **iptables** usa os conceitos de *chains (cadeias)* e *regras* para filtrar o tráfego.

O tráfego que entra no firewall e destinado ao próprio dispositivo de firewall é tratado pela chain **INPUT**. Exemplos desse tráfego são pacotes de ping provenientes de qualquer outro dispositivo em qualquer rede e enviados para qualquer uma das interfaces do firewall.

O tráfego originado no próprio dispositivo de firewall e destinado a outro lugar é tratado pela chain **OUTPUT**. Exemplos desse tráfego são respostas de ping geradas pelo próprio dispositivo de firewall.

O tráfego originou-se em outro lugar e a passagem pelo dispositivo de firewall é manipulada pela chain **FORWARD**. Exemplos desse tráfego são pacotes sendo roteados pelo firewall.

Cada chain pode ter seu próprio conjunto de regras independentes especificando como o tráfego deve ser filtrado para essa chain. Uma chain pode ter praticamente qualquer número de regras, incluindo nenhuma regra.

As regras são criadas para verificar características específicas dos pacotes, permitindo que os administradores criem filtros muito abrangentes. Se um pacote não corresponder a uma regra, o firewall passa para a próxima regra e verifica novamente. Se uma correspondência for encontrada, o firewall executará a ação definida na regra de correspondência. Se todas as regras em uma chain tiverem sido verificadas e ainda nenhuma correspondência foi encontrada, o firewall executará a ação especificada na política da chain, geralmente permite que o pacote flua ou negue.

a. Na **VM CyberOps Workstation**, inicie uma terceira janela de terminal R1.

```
mininet > xterm R1
```

b. Na nova janela do terminal **R1**, use o comando **iptables** para listar as chain e suas regras atualmente em uso:

```
[root @secOps ~] # iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target prot opt in out source destination

Chain FORWARD (policy ACCEPT 6 packets, 504 bytes)
 pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target prot opt in out source destination

[root @secOps ~] #
```

Quais chain estão atualmente em uso pelo **R1**?

c. Conexões com o servidor mal-intencionado geram pacotes que devem transverter o firewall **iptables** no **R1**. Os pacotes que atravessam o firewall são tratados pela regra FORWARD e, portanto, essa é a chain que receberá a regra de bloqueio. Para impedir que os computadores do usuário se conectem ao servidor mal-intencionado identificado na Etapa 1, adicione a seguinte regra à chain FORWARD no **R1**:

```
[root @secOps ~] # iptables -I FORWARD -p tcp -d 209.165.202.133 —dport 6666
-j DROP
[root @secOps ~] #
```

Onde:

- **-I FORWARD**: insere uma nova regra na chain FORWARD.
- **-p tcp**: especifica o protocolo TCP.
- **-d 209.165.202.133**: especifica o destino do pacote
- **—dport 6666**: especifica a porta de destino
- **-j DROP**: define a ação para soltar.

d. Use o comando **iptables** novamente para garantir que a regra foi adicionada à chain FORWARD. A VM CyberOps Workstation pode levar alguns segundos para gerar a saída:

```
[root @secOps analyst] # iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target prot opt in out source destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target prot opt in out source destination
    0 0 DROP tcp -- any any anywhere 209.165.202.133 tcp dpt:6666

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target prot opt in out source destination
[root@secOps analyst]#
```

e. Em **H5**, tente baixar o arquivo novamente:

```
[root @secOps analyst] # wget 209.165.202. 133:6666 /W32.Nimda.amm.exe
--2017-05-01 14:42:37-- http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... failed: Connection timed out.
Retrying.

--2017-05-01 14:44:47-- (try: 2) http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... failed: Connection timed out.
Retrying.
```

Digite **Ctrl+C** para cancelar o download, se necessário.

O download foi bem sucedido desta vez? Explique.

O que seria uma abordagem mais agressiva, mas também válida, ao bloquear o servidor ofensivo?

Parte 3: Terminar e limpar o processo Mininet

a. Navegue até o terminal usado para iniciar Mininet. Encerre o Mininet inserindo **quit** na janela principal do terminal da VM CyberOps.

b. Depois de sair da Mininet, limpe os processos iniciados pela Mininet. Entre a senha **cyberops** quando solicitado

```
[analyst@secOps scripts]$ sudo mn -c
[sudo] password for analyst:
```