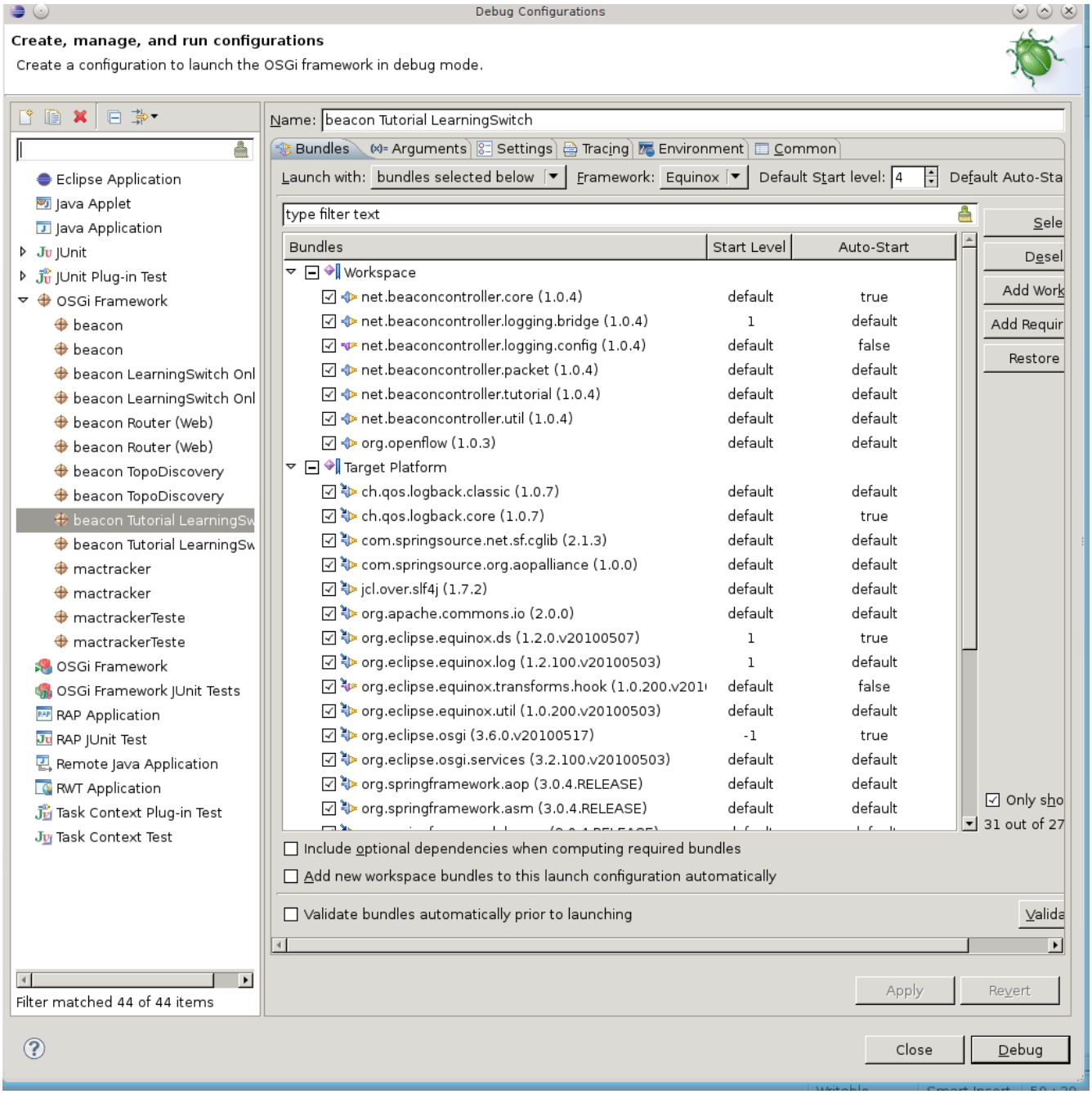
Of-IDPS

Passos para a execução do Of-IDPS:

1. Iniciar o controlador utilizando o *Eclipse/Run/DebugConfigurations*. Escolha "*beacon Tutorial LearningSwitch*" e clique em *debug*. Depois de iniciado o controlador escutará a porta 6633, tome cuidado para não ligar mais de um controlador ao mesmo tempo! Caso isso ocorra vai ocorrer um erro dizendo que a porta já está sendo utilizada, esse erro é bem comum. Caso o erro ocorra utilize o comando `netstat -a --tcp -np | grep 6633` para descobrir o processo da porta em execução e o comando `kill` para liberar a porta e execute o controlador novamente.



2. Iniciar a execução da máquina virtual com o mininet:
   a) Acessar via SSH a máquina virtual com o mininet. IP 192.168.1.200 usuário *mininet* senha *mininet*. O controlador deverá ter o IP 192.168.1.113. Comando:

      ```
      $ ssh -Y mininet@192.168.1.200
      ```

   b) Dentro da máquina virtual via SSH iniciar o ambiente que irá simular a rede com o switch OpenFlow. O controlador deve ter o IP 192.168.1.113. Comando (esse simula uma rede com um switch e três hosts):

      ```
      $ sudo mn --topo single,3 --mac --switch ovsk --controller
      remote,ip=192.168.1.113
      ```

   c) O passo anterior (b) ira iniciar o ambiente de controle do mininet, então é possível executar comandos diretamente desse console. Nesse iremos utilizar o comando `xterm` para executar um terminar para os hosts simulados pelo mininet. Nesse cenário de três máquinas execute o comando:

      ```
      $ xterm s1 h1 h2 h3
      ```

      Então será aberto um terminal para s1 (switch OF), host1, host2, host3.

      i. No terminar de s1 iremos habilitar o controle de banda (irá criar duas filas com larguras de bandas diferentes) e a espelhar uma porta do switch para colocar o IDS. Respectivamente os comandos:

         ```
         $./qos.sh
         $./mirror.sh
         ```

         Atenção! O comando `qos.sh` se adequá ao tamanho e quantidade de switches automaticamente. Mas o `mirror.sh` só é executado no switch s1 e com apenas três máquinas conectadas e o IDS deve obrigatoriamente ser o host 3 (h3)

      ii. No terminal do host3 (h3) inicie o IDS (snort).
         Se preciso apague as mensagens geradas anteriormente pelo snort com o comando:

         ```
         $  rm /var/log/snort/*
         ```

         Inicie o snort Comando:

         ```
         $  snort -c /etc/snort/snort.conf
         ```

3. No controlador: monte uma pasta entre o controlador e o computador com a mininet para compartilhar as mensagens geradas pelo IDS:

   ```
   $ sshfs mininet@192.168.1.200:/var/log/snort/ \
   /mnt/armazem/openflow/tmp/alertas/
   ```

4. Inicie o script que processa os alertas gerados pelo IDS na pasta compartilhada no passo 3. Acesse o diretório onde estão os arquivos de alerta que serão processados pelo controlador. Comando:

   ```
   $ cd /mnt/armazem/openflow/tmp/dadosSwitchesOF/
   ```
   Se desejado apague o arquivo existente:

   ```
   $ rm formatted_log.csv
   ```

   Então execute o script que ira a partir dos logs do snort gerar as mensagens de alerta no padrão que o controlador Of-IDPS reconhece com o comando:

   ```
   $ python snort_fast_alert_processor.py
   ```

   Observações 1 – Para executar o passo 4 é necessário que o passo 3 seja realizado com sucesso e que exista o arquivo `/mnt/armazem/openflow/tmp/alertas/alert.fast` que será processado pelo script do passo 4. Por fim, para que o controlador comece a processar os alertas é necessário que o passo 4 gere o arquivo `/mnt/armazem/openflow/tmp/dadosSwitchesOF/formatted_log.csv` que será consumido pelo software Of-IDPS do controlador.

   Observações 2 – As regras do IDS são enviadas para serem processadas pelo controlador através do script do passo 4. O Of-IDPS trabalha com o primeiro alerta que casar com o fluxo ganha, ou seja se temos primeiro um fluxo X com prioridade 3 que terá a sua largura de banda reduzida e depois um outro alerta para o mesmo fluxo X com prioridade 1 que será bloqueada o comportamento do Of-IDPS é aplicar a regra do primeiro e apenas do primeiro então esse fluxo X terá a sua largura de banda reduzida e não será bloqueado! Por isso, em testes pode ser interessante apagar as regras já existentes para ver o comportamento do Of-IDPS com as regras novas que serão geradas.

5. Após os passos anteriores o ambiente está pronto para ser testado! Recomendamos o `ping`, `iperf` e `idswakeup`. Para isso em abra um terminal no mininet com o comando xterm (ex. `xterm h1`) e por exemplo:

   a) Host 1 pingando host 2. comando aplicado em host 1:

      ```
      $ ping 10.0.0.2
      ```

      A rede 10.0.0.0/8 é o padrão do mininet, mas utilize os comandos comuns de rede Linux para verificar o seu ambiente, tal como `ifconfig`.

   b) Iperf, no host servidor, por exemplo host 1 execute:

      ```
      $ iperf -s -p 80
      ```

      No cliente, host 2:

      ```
      $ iperf -c 10.0.0.1 -p 80
      ```

c) Testando o IDS, do host 2 para host 1:

```
$ idswakeup 10.0.0.2 10.0.0.1 1 1
```

Resumo:

controlador/eclipse – F11 – Já escolhido "*beacon Tutorial LearningSwitch*"

```
controlador$ ssh -Y mininet@192.168.1.200

mininet$ sudo mn --topo single,3 --mac --switch ovsk --controller
remote,ip=192.168.1.113

mininet> xterm s1 h1 h2 h3

s1$ ./qos.sh
s1$ ./mirror.sh

h3$  rm /var/log/snort/*
h3$  snort -c /etc/snort/snort.conf

controlador$ sshfs mininet@192.168.1.200:/var/log/snort/ \
/mnt/armazem/openflow/tmp/alertas/

controlador$ cd /mnt/armazem/openflow/tmp/dadosSwitchesOF/
controlador$ rm formatted_log.csv
controlador$ python snort_fast_alert_processor.py
```

Depois disso utilize os terminais para testar o ambiente (passo 5).

Atenção comandos opcionais: h3$** `rm /var/log/snort/*` e **controlador$ `rm formatted_log.csv` cuidado **pagam/zeram** os alertas gerados anteriormente!