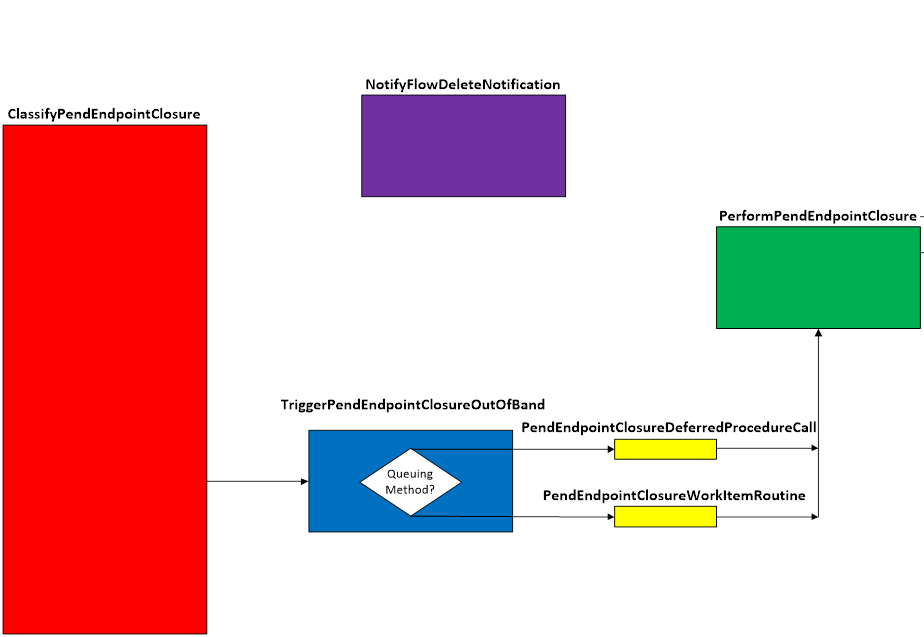
PEND ENDPOINT CLOSURE

**Overview**

The Pend Endpoint Closure scenario will cause the classification to pend for a specified period of time.  In a real world scenario, this time could be used to finish injecting outstanding NBLs.

All filters added sit in WFPSampler’s sublayer (which is weighted just below IPsec’s sublayer), unless otherwise specified using the –sl <SUBLAYER> command line option.  All filters are associated with WFPSampler’s provider.

The following diagram shows how the code flows for this callout:

  
**Figure A. Code flow for Pend Endpoint Closure Scenario**

When traffic matches a filter at the specified layer, **ClassifyPendEndpointClosure()** is invoked by the Filtering Engine.  This function will create the PEND\_DATA and call FwpsPendClassify().  If the classify has an appropriate flow associated with it (see FLOW\_ASOCIATION.mht), then the classify will exit with the expectation that the FlowDeleteFn will be invoked when the flow goes away.

Otherwise, **TriggerPendEndpointClosureOutOfBand()** is invoked.  Based on the queuing method, the appropriate queueFn is invoked.  The only time a DPC can be used is if there is no delay.  This is due to the fact that the delay introduced is done by using a function only available at PASSIVE\_LEVEL.  Introducing delays at DISPATCH\_LEVEL is rarely a good idea.

Regardless of which queueFn is used, each will call the **PerformPendEndpointClosure()**.

**PerformPendEndpointClosure()** will cause the thread to sleep for the duration provided.  When it wakes up, the classify is completed using FwpsCompleteClassify, and the PEND\_DATA is destroyed.

Note that with long delays, it is possible to bugcheck the machine.  This normally happens when a machine requests to drop into a power managed state.

**Applicable Layers**

❖  FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V4

❖  FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V6

**Command Line Usage**

|  |  |  |
| --- | --- | --- |
| **Option** | **Argument** | **Meaning** |
| -s | PEND\_ENDPOINT\_CLOSURE | Implement the PEND\_ENDPOINT\_CLOSURE scenario |
| -l | Applicable Layer | Layer at which this filter will apply |
| -pcd | Integer | How long of a pend completion delay to introduce (in ms) |
| -sl | Applicable subLayer | SubLayer to associate with the filter.  [default is WFPSAMPLER\_SUBLAYER]. |
| -v |  | Make the objects associated with this scenario’s instance dynamic |
| -b |  | Make the objects associated with this scenario’s instance available during boot-time |
| -tdpc |  | Use threaded DPCs for asynchronous (out of band) queuing method |
| -wi |  | Use work items for asynchronous (out of band) queuing method |
| -r |  | Remove objects associated with this scenario instance |
| -? |  | Display help |

“**WFPSampler.Exe -s PEND\_ENDPOINT\_CLOSURE -?**“ provides help output

“**WFPSampler.Exe -s PEND\_ENDPOINT\_CLOSURE -l FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V4 -v**“  adds a dynamic filter (**-v**) at FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V4 (**-l**) which references the appropriate callout.  This filter will have no conditions, meaning it will act on all closures seen at this layer.

“**WFPSampler.Exe -s PEND\_ENDPOINT\_CLOSURE -l FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V4  -v -r**“  removes (**-r**) the dynamic filter (**-v**) at FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V4 (**-l**) which references the appropriate callout.

“**WFPSampler.Exe -s PEND\_ENDPOINT\_CLOSURE -l FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V4 -ipla 1.0.0.1 -ipra 1.0.0.254 -pcd 5000**“ adds a persistent filter at FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V4 (**-l**) which references the appropriate callout.  This filter will have 2 conditions; FWPM\_CONDITION\_IP\_LOCAL\_ADDRESS (**-ipla**) equals 1.0.0.1, and FWPM\_CONDITION\_IP\_REMOTE\_ADDRESS (**-ipra**) equals 1.0.0.254. The closure will be delayed (**-pcd**) for 5 seconds.

“**WFPSampler.Exe -s PEND\_ENDPOINT\_CLOSURE -l FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V4  -aaid C:\Traffic.exe -ipla 1.0.0.1 -ipra 1.0.0.254 -ipp TCP -pcd 5000**“ adds a persistent filter at FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V4  (**-l**) which references the appropriate callout.  This filter will have 4 conditions; FWPM\_CONDITION\_ALE\_APP\_ID (**-aaid**) equals C:\Traffic.exe, FWPM\_CONDITION\_IP\_LOCAL\_ADDRESS (**-ipla**) equals 1.0.0.1, FWPM\_CONDITION\_IP\_REMOTE\_ADDRESS (**-ipra**) equals 1.0.0.254, and FWPM\_CONDITION\_IP\_PROTOCOL  (**-ipp**) equals TCP.  The classification will be delayed (**-pcd**) for 5 seconds.

For a list of conditions applicable to each layer, refer to Filtering Conditions Available at Each Filtering Layer.

For a list of command line parameters for configuring each condition, refer to Conditions for Command Line.

**Notes**

**FlowDelete**

Pending at FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V{4/6} can prevent flows from going away, which causes the FlowDeleteFn not to get invoked.  To work around this, you can call FwpsFlowRemoveContext on the context of the flow you need to  have the FlowDeleteFn invoked.  For a clear usage, look at ClassifyFunctions\_BasicStreamInjectionCallouts, and notice that if a flow was associated with both the STREAM and ALE\_ENDPOINT\_CLOSURE layers, a check is performed after the injection to determine if the flow is being terminated.  If it is, FwpsFlowRemoveContext is called on the flowContext for the ALE\_ENDPOINT\_CLOSURE, which will cause invocation of NotifyFlowDeleteNotification(), which will complete the pended classify.

**Mixing Scenarios**

PEND\_ENDPOINT\_CLOSURE is most useful when mixed with other scenarios.  For example:

                WFPSampler.exe -s BASIC\_STREAM\_INJECTION –l FWPM\_LAYER\_STREAM\_V4 -ipra 1.0.0.254 -iprp 6000 -v

                WFPSampler.exe -s PEND\_ENDPOINT\_CLOSURE -l FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V4 -aaid C:\Traffic.exe -ipra 1.0.0.254 -ipp TCP -iprp 6000 -pcd 5000 –v

                WFPSampler.exe -s FLOW\_ASSOCIATION –l FWPM\_LAYER\_ALE\_FLOW\_ESTABLISHED\_V4 -aaid C:\Traffic.exe -ipra 1.0.0.254 -ippTCP -iprp 6000 -aws BASIC\_STREAM\_INJECTION -awl FWPM\_LAYER\_STREAM\_V4 FWPM\_LAYER\_ALE\_ENDPOINT\_CLOSURE\_V4 –v

This will cause the endpoint associated with the specified flow to remain open for the duration of the injection being performed at STREAM.  Once the everything has been injected, the endpoint is allowed to close, and the flow contexts are cleaned up.