Suricata - Bug #1399

Task # 4380 (Assigned): tracking: improvements to bits, ints, vars

Flowbits rules not always evaluated in necessary order

02/26/2015 09:31 AM - David Wharton

Status:	Assigned		
Priority:	Normal		
Assignee:	Victor Julien		
Category:			
Target version:	TBD		
Affected Versions:		**Difficulty:**	
Effort:		**Label:**	

Description

Flowbits is a powerful and useful feature of Suricata and multiple flowbits rules can be applied to the same packet/stream. Unfortunately, there appears to be situations where flowbits rules are processed in an order that prevents them from alerting as expected.

Obviously, rules that set flowbits ("flowbit settters") have to be processed before rules that check flowbits ("flowbit checkers"). However, flowbits rules can both set and check flowbits and flowbits "chains" can exist and these rules must be evauated in a certain order based on the flowbits dependencies. Such dependencies do not appear to be respected in all cases by the Suricata engine when processing rules.

From what I can tell, when it comes to flowbits rules, rules that are only flowbit setters get processed first and rules that are only flowbit checkers get processed last. This makes sense and is appropriate. However, rules that both check and set flowbits seem to be proccessed in order based off of SID and this can create dependency issues for flowbits chains. (Note: this is based off my testing on version 2.0.6; older versions of Suricata like version 1.3.4 appear to base the processing order off of the order that the rules appear in the rules file.)

For example, consider the attached pcap. If it is run against the following ruleset that contains a simple flowbits chain, SIDs 1, 2, and 3 fire as expected:

```
alert tcp any any -> any any (msg:"First in chain"; content:"GET"; flowbits:set,1; sid:1;)
alert tcp any any -> any any (msg:"Second in chain"; content:"flow"; flowbits:isset,1; flowbits:se
t,2; sid:2;)
alert tcp any any -> any any (msg:"Third (Last) in chain"; content:"boss"; flowbits:isset,2; flowb
its:set,3; sid:3;)
```

However, if you change the SIDs in the chain like this, then only SIDs 2 and 3 fire:

```
alert tcp any any -> any any (msg:"First in chain"; content:"GET"; flowbits:set,1; sid:3;)
alert tcp any any -> any any (msg:"Second in chain"; content:"flow"; flowbits:isset,1; flowbits:se
t,2; sid:2;)
alert tcp any any -> any any (msg:"Third (Last) in chain"; content:"boss"; flowbits:isset,2; flowb
its:set,3; sid:1;)
```

The solution is, when there is a chain of flowbits, the rules need to be processed appropriately so that all necessary setters are processed before all dependent checkers. This may seem like a non-trivial task since complex dependent chains can be exist because rules can set and check more than one flowbit. Fortunately, there is a simple solution (at least in theory).

Flowbits rules can be represented as a directed acyclic graph (DAG) so the solution is nothing more than doing a topologial sort of them in order to determine appropriate processing order. (http://en.wikipedia.org/wiki/Topological_sorting)

If you view each rule as a node, and a directed edge exists from node A to node B if node A sets a flowbit checked by node B, then we end up with a DAG (unless there are loops in which case the graph wouldn't be acyclic and there would be a logical error in the ruleset preventing rules from alerting as expected; flowbits rules should always be able to be represented as a DAG).

In addition to determining rule processing order for flowbits rules, creating a graph of the rules provides other benefits:

1) Flowbits cycles (loops) can be detected and an error message can be thrown.
2) Rules that set flowbits that are never checked can be trivially identified and warning messages can be thrown.

History

#1 - 02/26/2015 10:57 AM - David Wharton

Also confirmed this behavior in version 2.0.7.

#2 - 09/08/2016 02:42 PM - Andreas Herz

- Assignee set to Anonymous

- Target version set to TBD

#3 - 11/23/2016 05:06 AM - Victor Julien

- Status changed from New to Assigned

- Assignee changed from Anonymous to Andreas Herz

- Target version changed from TBD to 70

Andreas can you try to reproduce this? Sigs should be sorted based on sets/issets, but maybe there is an issue with it. I vaguely recall fixing something around this though.

#4 - 11/23/2016 03:31 PM - Andreas Herz

Victor Julien wrote:

> Andreas can you try to reproduce this? Sigs should be sorted based on sets/issets, but maybe there is an issue with it. I vaguely recall fixing something around this though.

Yep I can still reproduce it with our recent master. This is the output from the run with just two alerts that trigger:

```
02/26/2015-14:11:13.136905  [**] [1:3:0] First in chain [**] [Classification: (null)] [Priority: 3] {TCP} 192.
168.114.3:13598 -> 172.19.113.215:80
02/26/2015-14:11:13.136905  [**] [1:2:0] Second in chain [**] [Classification: (null)] [Priority: 3] {TCP} 192
.168.114.3:13598 -> 172.19.113.215:80
```

So the last one is missing.

Do you have a hint what might have been the fix? Maybe it fixed some of that but not everything.

#5 - 11/24/2016 07:46 AM - Victor Julien

- Assignee changed from Andreas Herz to Victor Julien

Thanks Andreas, I'll have a look.

#6 - 07/09/2019 09:47 PM - Andreas Herz

This is still reproducible in 5.0 beta.

#7 - 08/07/2020 01:55 PM - Victor Julien

- Target version changed from 70 to TBD

#8 - 03/06/2021 08:07 AM - Victor Julien

- Parent task set to #4380

Files

suricata_flowbit_test.pcap	467 Bytes	02/26/2015	David Wharton