

Klever - Bug #8172

Unsafe verdicts of multimodule verification aren't associated with marks for single module verification

04/26/2017 12:17 PM - Alexey Polushkin

Status: New	Start date: 04/26/2017
Priority: High	Due date:
Assignee: Vladimir Gratinskiy	% Done: 0%
Category: Bridge	Estimated time: 0.00 hour
Target version:	Published in build:
Detected in build: svn	
Platform:	
Description After multimodule verification, existed marks don't apply to unsafe results. The reason for this is other names of verification object (say, <i>net/unix/unix.ko713c5ebc642b</i> instead of <i>net/unix/unix.ko</i>).	
Related issues: Related to Klever - Feature #8274: Support advanced attributes comparison New 07/05/2017	

History

#1 - 04/26/2017 12:56 PM - Evgeny Novikov

- Subject changed from *Unsafe verdicts of multimodule verification don't support existed marks* to *Unsafe verdicts of multimodule verification aren't associated with marks for single module verification*
- Description updated
- Assignee changed from *Alexey Polushkin* to *Vladimir Gratinskiy*
- Priority changed from *Normal* to *High*

We have the dedicated developer of Bridge that can do all the best. In particular here I suggest to always remove last 12 symbols of verification objects if they exactly match pattern [0-9a-f] whenever verification objects are compared, i.e. during marks association and verification jobs comparison (here all verdicts produced for one module will be merged and experts should analyze corresponding difference manually). This isn't applied for any attribute values, just for verification objects.

#2 - 04/26/2017 01:44 PM - Vladimir Gratinskiy

Evgeny Novikov wrote:

We have the dedicated developer of Bridge that can do all the best. In particular here I suggest to always remove last 12 symbols of verification objects if they exactly match pattern [0-9a-f] whenever verification objects are compared, i.e. during marks association and verification jobs comparison (here all verdicts produced for one module will be merged and experts should analyze corresponding difference manually). This isn't applied for any attribute values, just for verification objects.

I will not do something with attributes during marks associations as for safes associations I don't compare attributes values (since feature_8147 branch), just its ids. For unsafes I'm going to do the same. So I can fix attributes during reports uploading, but I guess its wrong. Right?

#3 - 04/26/2017 03:47 PM - Evgeny Novikov

Vladimir Gratinskiy wrote:

Evgeny Novikov wrote:

We have the dedicated developer of Bridge that can do all the best. In particular here I suggest to always remove last 12 symbols of verification objects if they exactly match pattern [0-9a-f] whenever verification objects are compared, i.e. during marks association and verification jobs comparison (here all verdicts produced for one module will be merged and experts should analyze corresponding difference manually). This isn't applied for any attribute values, just for verification objects.

I will not do something with attributes during marks associations as for safes associations I don't compare attributes values (since feature_8147 branch), just its ids. For unsafes I'm going to do the same. So I can fix attributes during reports uploading, but I guess its wrong. Right?

Actually all attribute values should be remained as is since this is exactly what users expect to see. But here we encounter a new conception - there are attribute values for visualization and attribute values for comparison. So, I suggest the following. Let's instead of a field for attributes use two

fields: the first one visualization attributes (they are exactly equal to report attributes) and the second one comparison attributes. You can fill both these fields when reports are uploaded using corresponding rules (I assume that in the most cases both fields will keep the same values). Then you should use an appropriate field when you, say, show report attributes or associate marks.

Since rules for translating visualization attributes into comparison ones can change there should be an ability to recalculate comparison attributes and everything related, e.g. mark associations. I am not sure that this will be very useful but I am sure that this will be quite hard, so this can be missed right now.

BTW, don't think that all rules will be like simple regular expressions. For instance, we already met many times situations when module was renamed in a newer version of the Linux kernel so straightforward comparison and mark associations fail. That's why one would like to specify by hand such the rule that although this module will be shown as *mod-new.ko* it will be compared and associated with marks as *mod-old.ko*. And so on. After all rules can become quite complicated.

Moreover, in accordance with a generic method suggested by me and Ilja in our paper Bridge should allow to configure various things like mentioned rules via interface rather than via changing its source files. But this is another story for which I will open a new issue.