

Re: [AppArmor 39/45] AppArmor: Profile loading and manipulation, pathname matching

Re: [AppArmor 39/45] AppArmor: Profile loading and manipulation, pathname matching

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-06/msg04286.html>

- *From:* Stephen Smalley <sds@xxxxxxxxxxxxxx>
 - *Date:* Mon, 11 Jun 2007 11:16:05 -0400
-

On Sat, 2007-06-09 at 00:03 +0200, Andreas Gruenbacher wrote:

On Wednesday 06 June 2007 15:26, Stephen Smalley wrote:

– under AA, each file may have an arbitrary set of "labels" or "policies" applied to it depending on what programs are accessing it and what names are being used to reference it – there is no system view of the subjects and objects and thus no way to identify the overall system policy for a given file.

Look at it this way: under SELinux, the set of files that share a label forms an equivalence class — they are all treated identically by the system's security policy. The rules in AppArmor profiles also define equivalence classes in the sense that they partition the filesystem namespace into sets of files that are treated identically, but this classification is not explicit — the entire rule base contributes to the classification. This doesn't mean that there is not a global policy, just that the policy is modeled differently. The equivalence classes are not directly obvious from the AA profiles.

No, it really does mean that there is no global policy, and it goes beyond "not directly obvious" to "can not be determined" from the AA profiles. You can't compose the set of AA profiles and say anything useful, because they are written in terms of ambiguous and unstable identifiers. /a/b/c may refer to completely different objects in two different profiles, or to the same object as /d/e/f in the same or another profile.

Contrast this with SEEdit, which compiles AA-style rules into labels (and thus equivalence classes). The resulting SELinux policy is a static snapshot that cannot easily accommodate rule base changes, is more limited with respect to new files (which would likely be fixable), and behaves differently in complex ways with file renames. What's more, most likely the compiled policy will be anywhere from very hard to impossible to analyze, so you pretty much lose on all ends.

Re: [AppArmor 39/45] AppArmor: Profile loading and manipulation, pathname matching

Re: [AppArmor 39/45] AppArmor: Profile loading and manipulation, pathname matching

Just to clarify, you can change the allowed accesses from a given subject to a given object without relabeling, just by changing the policy allow rules; you only have to relabel the object in the case where you want to distinguish that object from another object with the same label for the same subject. I think the new file situation could be improved without any major change to the SELinux model, and am not opposed to leveraging the component name there, as previously noted. On the file rename case, I think we have it right – access rights shouldn't change automatically when a file is renamed, any more than DAC ownership or file modes should.

– names are far less tranquil than labels.

If I'm getting things right, a tranquil system with respect to labels would be one that does not permit re-labeling, while a tranquil system with respect to path names would be one that does not permit renaming. Both approaches would buy greater analyzability with reduced usability, and both seem unrealistic to me. SELinux and AppArmor evidently have different goals, and tranquility is more important to SELinux.

Tranquility is important to correctness and understandability of policy; if labels (or pathnames in your case) can change at any time, then you have the problems of revocation of access (impractical to completely implement in Linux) and your effective policy now varies over time, so you have to consider time as a factor in your policy analysis.

AppArmor is meant to be relatively easy to understand, manage, and customize, and introducing a labels layer wouldn't help these goals. SELinux is applicable in areas where AppArmor is not (e.g., MLS), but this comes at a cost. For me the question is not SELinux or AppArmor, but if AppArmor's security model is a good solution in common scenarios. In my opinion, AppArmor is a better answer than SELinux in a number of scenarios. This gives it value, notwithstanding the fact that SELinux can be taken further.

I'd agree that we shouldn't try to emulate AA as it is on SELinux. The question is more of whether we can meet the higher level functionality goals that make some people want to use AA via SELinux. That requires separating those goals from the implementation details of AA.

—

Stephen Smalley
National Security Agency

–

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in

Re: [AppArmor 39/45] AppArmor: Profile loading and manipulation, pathname matching

Re: [AppArmor 39/45] AppArmor: Profile loading and manipulation, pathname matching

the body of a message to majordomo@xxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>