Closed Issue created 1 year ago by Vantler

Selinux MCS generate a single category context and may be accessed by another machine

Goal

A machine(image file) with context like system_u:system_r:svirt_tcg_t:s0:c423 can be accessed by a machine with context like system_u:system_r:svirt_tcg_t:s0:c423,c792. This should be avoided.

Technical details

In src/security/security_selinux.c, virSecuritySELinuxMCSFind(), We can see that the program randomly gets two numbers. But if c1 == c2, the program will generate a single category context like s0:cXXX,

```
if (c1 == c2) {
            mcs = g_strdup_printf("%s:c%d", sens, catMin + c1);
        }
```

But if we have got machine with context like "s0:cXXX,cYYY" ,It will be able to read the image of machine with "s0:cXXX". This should be avoided.

Additional information

```
if (c1 == c2) {
            VIR_FREE(mcs);
            continue;
        }
```

To upload designs, you'll need to enable LFS and have an admin enable hashed storage. More information

Tasks 0

No tasks are currently assigned. Use tasks to break down this issue into smaller parts.

Linked items 0

Link issues together to show that they're related or that one is blocking others. Learn more.

Activity

Peter Krempa added kind bug scoped label 1 year ago

Peter Krempa added security-selinux label 1 year ago

Daniel P. Berrangé @berrange · 1 year ago Owner

I have confirmed this behaviour and it does indeed appear to be insecure. AFAICT, a file can be accessed with its set of MCS categories is equal-to, or a subset-of, the MCS categories of the process. eg A vm running

```
  svirt_t:s0:c117,c720,c890
```

is able to access files labelled with any of

```
  svirt_image_t:s0:c117
  svirt_image_t:s0:c720
  svirt_image_t:s0:c890
  svirt_image_t:s0:c117,c720
  svirt_image_t:s0:c720,c890
  svirt_image_t:s0:c117,c890
  svirt_image_t:s0:c117,c720,c890
```

Considering the 2 category case that libvirt uses for label generation. If we have a range of MCS categories 0-1023, we have `1024*1024` combinations, but we only accept ordered pairs, so it is more like `1024*1024/2` . If there are 1024 cases where c1==c2, then the probability of having a VM with a single MCS category is approx 0.2% (calc: `1024/(1024*(1024+1)/2)*100`). Luckily this is small enough that the impact of this bug is quite minor, at least for moderate VM counts. It is reduced further if we were to add in probability of having another VM on the same host with a category pair, one of whose categories matches.

None the less it is clearly critical to fix.

Edited by Daniel P. Berrangé 1 year ago

Gianluca Gabrielli @crazybyte · 1 year ago

This issue got `CVE-2021-3631` assigned.

Daniel P. Berrangé closed via commit 15073504 1 year ago

Zqiang1211 mentioned in commit distro-poky/layers/meta-virtualization@0644e808 1 year ago

Fabrice Fontaine mentioned in commit buildroot.org/buildroot@93cbbb2c 1 year ago

Fabrice Fontaine mentioned in commit cronmod-dev/buildroot@5792946e 1 year ago

Please register or sign in to reply