Closed · Issue created 2 years ago by **Aviv Sasson**

Security Disclosure

Hi, I've found a vulnerability in Slirp and would like to disclose it. In order to make the disclosure secure please contact me on my mail - asasson@paloaltonetworks.com @elmarco

Drag your designs here or click to upload.

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. Learn more.

Related merge requests 1

Cve 2020 1983
!38

When this merge request is accepted, this issue will be closed automatically.

Activity

Marc-André Lureau @elmarco · 2 years ago — Owner

Hi @asasson , this issue is confidential, so feel free to report it here. Have you also started a CVE?

Aviv Sasson @asasson · 2 years ago — Author

Alright great,

The vulnerability is here -

```
#2 0x55ad4ff83159 in m_cat vendor/libslirp/src/mbuf.c:133
#3 0x55ad4ff809c0 in ip_reass vendor/libslirp/src/ip_input.c:337
#4 0x55ad4ff7f80e in ip_input vendor/libslirp/src/ip_input.c:184
#5 0x55ad4ff446a4 in slirp_input vendor/libslirp/src/slirp.c:835
#6 0x55ad4ff3bcf0 in do_slirp /home/osboxes/slirp4netns-project/slirp4netns-test_crush_etc_aviv/slirp4netns.c:337
#7 0x55ad4ff37a97 in parent /home/osboxes/slirp4netns-project/slirp4netns-test_crush_etc_aviv/main.c:259
#8 0x55ad4ff3a27d in main /home/osboxes/slirp4netns-project/slirp4netns-test_crush_etc_aviv/main.c:665
#9 0x7f3faaf4bb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
```

The vulnerability is a use after free that cause a denial of service.

In ip_reass(), when sending 2 ip packets that are supposed to fragment into one. The problem is that if the length of those 2 packets together is bigger than 65K then the program crush.

When it happened, libslirp realloc one of the mbufs (ip_input.c:337) and then when ip_reass returns it uses the old reference of the mbuf and collapse.

Attached are 2 packets and a Scapy script that exploit the issue. In addition, in order to make it work you should delete the cksum check in ip_input() because the cksum of those packet is incorrect. (Sorry for the trouble, I didn't had much time to create a full functioning PoC)

I've reserved CVE-2020-1983 for the issue, Let me know if you need more information regarding the issue.

Aviv Sasson Sr. security researcher, Palo Alto Networks

```
from scapy.all import *

a = open("212","rb")

b = a.read(1000000)

buff1 = b[14:]

t = IP(buff1)

t.version = 4

send(t)

a = open("214","rb")

b = a.read(1000000)

buff1 = b[14:]

t = IP(buff1)

t.version = 4

send(t)
```

212 214

Edited by Marc-André Lureau 2 years ago

Marc-André Lureau @elmarco · 2 years ago — Owner

Your IP packets are really big, over 64k, I can't send them over tun. How did you setup the guest/container network?

Marc-André Lureau @elmarco · 2 years ago — Owner

@asasson ^

Marc-André Lureau @elmarco · 2 years ago — Owner

I think what you describe was fixed by commit 126c04ac.

(But ip_reass() still looks fishy - so I won't be surprised to find more issues)

Aviv Sasson @asasson · 2 years ago — Author

Hi, My setup is Podman. It's a container platform that uses slirp. The command "podman run --log-level debug -it --rm docker.io/library/ubuntu /bin/sh" will give you a shell on a container and over there you can run the script and see that slirp4netns(uses libslirp) crush.

Please register or sign in to reply

Samuel Thibault @sthibaul · 2 years ago — Owner

commit 126c04ac only fixes the case when we switch from the preallocated buffer to a dynamically-allocated buffer. It doesn't fix the case when then `m_inc()` call inside `m_cat()` extends an already-dynamically-allocated buffer.

Marc-André Lureau @elmarco · 2 years ago — Owner

oh I see, thanks

Marc-André Lureau @elmarco · 2 years ago — Owner

So that should do it?

```
diff --git a/src/ip_input.c b/src/ip_input.c
index aa514ae..9bbcd1f 100644
--- a/src/ip_input.c
+++ b/src/ip_input.c
@@ -328,8 +328,6 @@ insert:
      q = fp->frag_link.next;
      m = dtom(slirp, q);

-     int was_ext = m->m_flags & M_EXT;
-
      q = (struct ipasfrag *)q->ipf_next;
```

```
    while (q != (struct ipasfrag *)&fp->frag_link) {
        struct mbuf *t = dtom(slirp, q);
@@ -351,7 +349,7 @@ insert:
     * then an m_ext buffer was alloced. But fp->ipq_next points to the old
     * buffer (in the mbuf), so we must point ip into the new buffer.
     */
-    if (!was_ext && m->m_flags & M_EXT) {
+    if (m->m_flags & M_EXT) {
        int delta = (char *)q - m->m_dat;
        q = (struct ipasfrag *)(m->m_ext + delta);
    }
```

Please register or sign in to reply

Samuel Thibault @sthibaul · 2 years ago — Owner

no, because `q - m->m_dat` doesn't make sense when was_ext is true

Edited by Samuel Thibault 2 years ago

Marc-André Lureau @elmarco · 2 years ago — Owner

and m_data either I suppose. gosh that code is #@&@!.. Are you working on a fix?

Edited by Marc-André Lureau 2 years ago

Please register or sign in to reply

Samuel Thibault @sthibaul · 2 years ago — Owner

`m->m_data` before the `m_inc` should be fine. Possibly just saving it before, and using that instead of `m->m_dat` (in addition to your change) would just work.

Marc-André Lureau @elmarco · 2 years ago — Owner

m->m_data doesn't point to start of alloc region. Other proposal:

```
diff --git a/src/ip_input.c b/src/ip_input.c
index aa514ae..a092ba9 100644
--- a/src/ip_input.c
+++ b/src/ip_input.c
@@ -326,10 +326,9 @@ insert:
     * Reassembly is complete; concatenate fragments.
     */
    q = fp->frag_link.next;
    m = dtom(slirp, q);
+   int delta = (char *)q - (m->m_flags & M_EXT ? m->m_ext : m->m_dat);

-   int was_ext = m->m_flags & M_EXT;
-
    q = (struct ipasfrag *)q->ipf_next;
    while (q != (struct ipasfrag *)&fp->frag_link) {
        struct mbuf *t = dtom(slirp, q);
@@ -351,8 +350,7 @@ insert:
     * then an m_ext buffer was alloced. But fp->ipq_next points to the old
     * buffer (in the mbuf), so we must point ip into the new buffer.
     */
-   if (!was_ext && m->m_flags & M_EXT) {
-       int delta = (char *)q - m->m_dat;
+   if (m->m_flags & M_EXT) {
        q = (struct ipasfrag *)(m->m_ext + delta);
    }
```

Edited by Marc-André Lureau 2 years ago

Please register or sign in to reply

Samuel Thibault @sthibaul · 2 years ago — Owner

That could work indeed.

Aviv Sasson @asasson · 2 years ago — Author

Hi, it looks good, when I'm running the software with the fix it doesn't crush.

Edited by Aviv Sasson 2 years ago

Marc-André Lureau @elmarco · 2 years ago — Owner

thanks @asasson. Last time we had a CVE, and handled a confidential issue, we simply merged the fix and let distros do backport.

Is that fine? How do you handle the CVE? Can you update it to reference a commit fix?

Edited by Marc-André Lureau 2 years ago

Marc-André Lureau @elmarco · 2 years ago — Owner

I don't know how to make a confidential MR.

@sthibaul, can you review?: https://gitlab.freedesktop.org/slirp/libslirp-private/-/commit/83eb7c66fef7f236a6ffd75902f5c939336c9d90

Aviv Sasson @asasson · 2 years ago — Author

Sounds like a good plan. @elmarco Once the fix will be public I'll release the CVE and add a reference to the commit and to whatever you'll like.

Marc-André Lureau mentioned in merge request !38 (merged) 2 years ago

Marc-André Lureau assigned to @elmarco 2 years ago

Marc-André Lureau closed via merge request !38 (merged) 2 years ago

Marc-André Lureau mentioned in commit 9ac0371b 2 years ago

Marc-André Lureau made the issue visible to everyone 2 years ago

Akihiro Suda @AkihiroSuda · 2 years ago — Contributor

@elmarco @sthibaul Could you ship a new release?

Marc-André Lureau @elmarco · 2 years ago — Owner

@AkihiroSuda master should be fine for a release. @sthibaul ok?

Samuel Thibault @sthibaul · 2 years ago — Owner

Ok for me

Please register or sign in to reply