New issue

Jump to bottom

isisd: overflow bugs in unpack_tlv_router_cap #10507

Closed whichbug opened this issue on Feb 5 · 21 comments

whichbug commented on Feb 5 • edited

frr/isisd/isis_tlvs.c
Lines 3004 to 3114 in 18ed776

```
3004      subtlv_len = tlv_len - ISIS_ROUTER_CAP_SIZE;
3005      while (subtlv_len > 2) {
3006              uint8_t msd_type;
3007
3008              type = stream_getc(s);
3009              length = stream_getc(s);
3010              switch (type) {
3011              case ISIS_SUBTLV_SID_LABEL_RANGE:
3012                      /* Check that SRGB is correctly formated */
3013                      if (length < SUBTLV_RANGE_LABEL_SIZE
```

There are a few issues in the loop above leading to overflow vulnerabilities. The loop condition is `subtlv_len > 2` and `subtlv_len` is updated at the end of the loop (Line 3113: `subtlv_len = subtlv_len - length - 2`).

The Issue on Loop Condition: at Line 3113, when we update `subtlv_len`, if `subtlv_len < length + 2`, integer overflow will happen, leading to heap overflows in the next loop iteration. Note that `subtlv_len` and `length` are of `uint8_t`, which makes it easy to overflow, e.g., `subtlv_len=35`, `length=38` ... An example output of the address sanitizer can be found below:

```
2022/02/05 21:54:53 ISIS: [S48XV-GT1WC][EC 100663311] stream_getc: Attempt to get char out of
bounds
2022/02/05 21:54:53 ISIS: [RF6RD-ZCBTC][EC 100663311] &(struct stream): 0x61200000f940, size: 231,
getp: 74, endp: 74
==86437==
    #1 0x7fe6ec3a118a in __libc_signal_restore_set /build/glibc-eX1tMB/glibc-
2.31/signal/../sysdeps/unix/sysv/linux/internal-signals.h:86:3
    #2 0x7fe6ec3a118a in raise /build/glibc-eX1tMB/glibc-
2.31/signal/../sysdeps/unix/sysv/linux/raise.c:48:3
```

```
#3 0x7fe6ec380858 in abort /build/glibc-eX1tMB/glibc-2.31/stdlib/abort.c:79:7
#4 0x880f5c in _zlog_assert_failed /home/parallels/myfrr/lib/zlog.c:700:2
#5 0x7f2f6f in stream_getc /home/parallels/myfrr/lib/stream.c
#6 0x63c10c in unpack_tlv_router_cap /home/parallels/myfrr/isisd/isis_tlvs.c:3016:10
#7 0x6242a5 in unpack_tlv /home/parallels/myfrr/isisd/isis_tlvs.c:4332:10
#8 0x6242a5 in unpack_tlvs /home/parallels/myfrr/isisd/isis_tlvs.c:4354:8
#9 0x623ce4 in isis_unpack_tlvs /home/parallels/myfrr/isisd/isis_tlvs.c:4385:7
#10 0x5d5b5b in process_lsp /home/parallels/myfrr/isisd/isis_pdu.c:940:6
#11 0x5cf009 in isis_handle_pdu /home/parallels/myfrr/isisd/isis_pdu.c:1781:12
```

Other Issues: at Line 3016, Line 3021, Line 3062, Line 3067, and Line 3098, I think we need to use `break` instead of `continue`. Using `continue` will let us miss the update of the loop condition variable at Line 3113.

Please check if my understanding of the code above is correct. If so, I can make a pull request to fix these issues then.

whichbug commented on Feb 5 — Author

Issues on stream_forward_getp: Also in the loop shown above, at Line 3015, Line 3020, Line 3045, Line 3054, Line 3061, Line 3066, Line 3097, Line 3107, and Line 3110, when we call `stream_forward_getp(s, length)`, there is no check whether `s + length` exceeds the bound of the stream, thus causing assertion failures in `stream_forward_getp`.

whichbug commented on Feb 5 — Author

Issues on stream_get: Also in the loop shown above, at Line 3049, when we call `stream_get`, we do not check if enough data are available in the buffer, which will cause overflows...

whichbug commented on Feb 5 • edited — Author

Integer overflow bugs: Also in the loop shown above, at Line 3043 and Line 3090, when we do the operation `size = length - (size + SUBTLV_SR_BLOCK_SIZE)`, integer overflows often happens, when `length < size + SUBTLV_SR_BLOCK_SIZE`. Both `length` and `size` are of uint8_t, making the overflow happen frequently...

And I think the following code at Lines 3045 and 3092 should be `stream_forward_getp(s, size)` rather than `stream_forward_getp(s, length)`. Please check...

whichbug commented on Feb 5 • edited — Author

Heap overflows: Also in the loop shown above, at Line 3031 and Line 3078, the operation `stream_getl(s)` is valid only when `length = 10` but `length` may be `9` ...

qlyoung commented on Feb 7 — Member

Hi! Are you fuzzing or auditing manually?

whichbug commented on Feb 7 — Author

> Hi! Are you fuzzing or auditing manually?

Hi, I am using FRR in one of our projects and trying to test its reliability. You can regard it as a kind of testing.

I carefully went through every problem I found and reported the issue only when I thought it is really a problem. I also tried to propose fixes as I also want to make contributions to FRR's community.

qlyoung commented on Feb 7 — Member

Yes, your reports are good quality and appreciated. I'm just curious what techniques you're using to find the issues. Are you part of a group? Others seem to be reporting the same type of issues recently.

idryzhov commented on Feb 7 — Contributor

Please check PR #9850.
I believe most of the reported issues are solved there. Or even all of them.
The PR is finished and the code is approved, but unfortunately it breaks some of our tests so it was not merged.
I believe the tests could be wrong and should be fixed so it would be great if you look into that.

idryzhov commented on Feb 7 — Contributor

Update: #9850 has conflicts with current master, so I pushed a rebased version in #10517. Let's see whether it passes the tests.

whichbug commented on Feb 7 — Author

> Update: #9850 has conflicts with current master, so I pushed a rebased version in #10517. Let's see whether it passes the tests.

Thanks very much. That patch is helpful for me.

idryzhov commented on Feb 8 — Contributor

#10517 is now merged, could you please check if it fixes all reported issues?

whichbug commented on Feb 8 — Author

> #10517 is now merged, could you please check if it fixes all reported issues?

Yes, I think all are fixed. This issue can be closed. Thanks for your great efforts!

idryzhov commented on Feb 8 — Contributor

Thanks for reporting!

 **idryzhov** closed this as completed on Feb 8

00xc commented on Feb 25

This was assigned CVE-2022-26125.

qlyoung commented on Mar 25 — Member

Yeah, so...this is an assertion failure unless I'm reading the issue wrong. We push a stream buffer until a point where automatic bounds checks kick in and terminate the program to *prevent* a buffer overflow. Yet this is labeled as a buffer overflow and assigned a CVE score of 7.8.

@00xc @whichbug can one of you guys clarify the situation here?

whichbug commented on Mar 25 — Author

> Yeah, so...this is an assertion failure unless I'm reading the issue wrong. We push a stream buffer until a point where automatic bounds checks kick in and terminate the program to *prevent* a buffer overflow. Yet this is labeled as a buffer overflow and assigned a CVE score of 7.8.
>
> **@00xc @whichbug** can one of you guys clarify the situation here?

I guess the problem is that the assertion failures are caused by potential overflows. The CVE score could be reduced.

qlyoung commented on Mar 28 — Member

There are no potential overflows. The potential for overflow is eliminated by the assert. That's the whole point of the assert, that's why we put it there.

Since it sounds like you agree with my assessment, can you work with the CVE people to fix the CVE you filed? Frankly I don't even think this deserves a CVE but if you are set on filing them, please make sure you accurately report the problem.

So that you have some insight into what happens when the CVE process is used, we now have people from Debian emailing us regarding the high-severity CVEs you've filed for these issues. So now that I've found that this one isn't accurate, now I have to go through and cross check each of your reports against the CVEs you've filed to make sure they're actually accurate.

On that topic, usually when you file CVEs against projects, it's expected that you work with the people who run the project to agree on a characterization of what is going to be filed and when it's going to be disclosed. Unless we missed some communication from you, you haven't done that, which is inconsiderate to us. We're volunteers, not a major company selling a product. We aren't going to sue you to prevent you from filing CVEs. We have an established process for reporting security vulnerabilities that's in the project README. Your research is appreciated but please, follow common practices, spend the time to reach out and work with us regarding reporting and disclosure.

This was referenced on Mar 28

isisd: misusing strdup leads to stack overflow #10505

Closed

Miss a check on length in Babel #10487

Closed

Incorrect checks on length in babeld #10502

Closed

babeld: bugs in parse_hello_subtlv, parse_ihu_subtlv, and parse_update_subtlv #10503

Closed

qlyoung commented on Mar 28 — Member

@00xc @whichbug By the way, if you guys are really interested in helping us with project security, we have a big list of issues just like this one found via our own security processes (such as Coverity, LLVM scan-build, ClusterFuzz, etc) that we need help fixing. Let me know if you're interested in contributing and I'll get you connected with our services so we can get some patches rolling.

whichbug commented on Mar 28 — Author

> **@00xc @whichbug** By the way, if you guys are really interested in helping us with project security, we have a big list of issues just like this one found via our own security processes (such as Coverity, LLVM scan-build, ClusterFuzz, etc) that we need help fixing. Let me know if you're interested in contributing and I'll get you connected with our services so we can get some patches rolling.

Thanks for your information. It looks good to me if I can contribute. You see I always tried to fix these reported bugs.

whichbug commented on Mar 28 Author

> Since it sounds like you agree with my assessment, can you work with the CVE people to fix the CVE you filed?

I have sent a request to update. I think it will be updated soon.

00xc commented on Mar 29

> On that topic, usually when you file CVEs against projects, it's expected that you work with the people who run the project to agree on a characterization of what is going to be filed and when it's going to be disclosed.

I don't think this was directed to me, but to be clear, I did not request the CVE, I just tried to make everyone aware. For what is worth, I updated our downstream CVSS score for this issue accordingly. Hopefully MITRE will do the same soon.

> Frankly I don't even think this deserves a CVE

Not that my personal opinion matters much here, but a remotely reachable assertion (i.e. remote DoS) seems severe enough, although I might be missing something.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

No milestone

Development

No branches or pull requests

4 participants