Telekom **Security**

Wire Secure Messenger Remote Format String Vulnerability

09 Nov 2020

A Remote Format String Vulnerability in the Wire Secure Messenger (CVE-2020-27853) allows an attacker to cause a denial of service (application crash) or possibly execute arbitrary code via voice or video call. This affects Wire AVS (Audio, Video, and Signaling) 5.3 through 6.x before 6.4, the Wire Secure Messenger application before 3.49.918 for Android, and the Wire Secure Messenger application before 3.61 for iOS.

CVE ID:	**CVE-2020-27853**
Affected Products:	**Wire AVS (Audio, Video, and Signaling) 5.3 through 6.x before 6.4** **Wire Secure Messenger Android before 3.49.918** **Wire Secure Messenger iOS before 3.6.1**
Risk:	**Critical**
Discovered by:	**Roman Wagner, Holger Fuhrmannek**

How the vulnerability was found?

During a penetration test of the Wire Secure Messenger different remote attack vectors were investigated. As a result of this, the processing of SDP (Session Descripton Protocol) packets was analyzed. SDP is used as a typical session setup (exchange of specifications and capabilities) before a media stream (e.g. audio) is transferred between peers. This process is called Signaling. The open source library Wire AVS (Audio, Video and Signaling) is responsible for handling those SDP packets. During the further investigation the library AVS was analyzed with a fuzzing approach.

How the library Wire AVS was fuzzed?

A code coverage based fuzzing approach (AFL and Libfuzzer) was selected. In addition, the AddressSanitizer was used to promote the detection of heap and stack vulnerabilities during the fuzzing assessment. Interesting AVS functions that are related to the processing of SDP packets were selected. The dynamic instrumentation tool Frida was used to verify that the selected AVS functions were called during a Wire voice or video call.

How the vulnerability is triggered?

The vulnerability is triggered by the parsing of the SDP description. A snippet of a SDP description that was recorded during the assessment is illustrated in the following:

```
v=0
o=- 4195135229 1485794652 IN IP4 127.0.0.1
s=-
c=IN IP4 127.0.0.1
t=0 0
a=group:BUNDLE 0 1 2
a=msid-semantic: WMS 2367a44e-4060-4f8e-8a50-b3cfc793c996 a=tool:avs 5.5.10 (arm64)
m=audio 49126 UDP/TLS/RTP/SAVPF 111
a=rtpmap:111 opus/48000/2
a=fmtp:111 minptime=10;useinbandfec=1
...
```

The processing of SDP session attributes “a=key:value” could trigger a format string vulnerability that is exploitable by an attacker.

The function call `sdp_media_set_lattr()` in src/peerflow/sdp.c of Wire AVS is triggering the vulnerability.

```
// library AVS, sdp.c:
static bool media_rattr_handler(const char *name, const char *value, void *arg){
        ...
    sdp_media_set_lattr(sdpm, false, name, value);
        ...
}
```

The function parameter **value** of `sdp_media_set_lattr()` is derrived from the SDP session attribute and can be replaced with a format string such as “%s” by an attacker. The function `sdp_media_set_lattr()` is implemented in the imported library re.

```
// library re, media.c:
int sdp_media_set_lattr(struct sdp_media *m, bool replace, const char *name, const char *value, ...){
        ...
        err = sdp_attr_addv(&m->lattrl, name, value, ap);
        ...
}

// library re, attr.c
int sdp_attr_addv(struct list *lst, const char *name, const char *val, va_list ap){
        ...
        err |= re_vsdprintf(&attr->val, val, ap);
        ...
}
```

The SDP session attribute is forwarded as second parameter to the function `re_vsdprintf()` where the vulnerability is triggered.

Impact

An externally-controlled format string could lead to buffer overflows, denial of service, or data representation problems. During the fuzzing multiple payloads were identified that lead to a denial of service of the Wire application. Moreover, the AddressSanitizer showed that some payloads were able to trigger heap overflows.

To exploit the vulnerability an attacker need to start a voice or video call to a contact and replace the SDP attribute value during the Signaling with a malformed format string. If the victim accepts the call, the format string vulnerability is triggered. This leads at least to a denial of service (application crash) and potentially to Remote Code Execution.

References

- https://nvd.nist.gov/vuln/detail/CVE-2020-27853
- https://github.com/wireapp/wire-audio-video-signaling/issues/23
- https://wire.com/
- https://llvm.org/docs/LibFuzzer.html
- https://lcamtuf.coredump.cx/afl/

Roman Wagner (roman.wagner@t-systems.com**)**