Publication: ZD Nets Online
Date: 01 October 2019
Headline: Academics find eight vulnerabilities in Android's VoIP components

Academics find eight vulnerabilities in Android's VoIP components

A team of academics has found eight vulnerabilities in the Android operating system's VoIP components. These vulnerabilities can be exploited to make unauthorized VoIP calls, spoof caller IDs, deny voice calls, and even execute malicious code on users' devices.

The research is the first of its kind. Until now, security researchers and academics have only looked at the security features of Voice-over-IP (VoIP) equipment, servers, and VoIP mobile apps, but none have analyzed the VoIP components inside Android itself.

The three-man research team set out to correct this. Over the course of the past few years, they developed three methods of analyzing Android's VoIP backend and systematically combed through the components for security flaws that could be exploited by an attacker.

Most of their testing revolved around using fuzzing, a well-known automated software testing technique that relies on blasting random and malformed data into a software component and observing how it reacts, looking for abnormalities in the output, such as crashes or memory leaks.

The research team said they first fuzzed the Android Intent and System APIs for interactions with the operating system's native VoIP components; they then set up a VoIP testbed in their laboratory and fuzzed the different VoIP protocols (such as SIP [Session Initiation Protocol], SDP [Session Description Protocol], and RTP [Real-time Transport Protocol]); and then they manually reviewed logs and performed manual code audits, as the last stage of their research.

They only tested recent versions of the Android OS, from Android 7.0 (Nougat) to last year's 9.0 (Pie). The research team's work produced nine vulnerabilities, all of which were reported to Google, and some fixed. Of the nine, eight impacted Android's VoIP abckend, while a ninth (V1 in the table below), impacted a third-party app.

Publication: ZD Nets Online
Date: 01 October 2019
Headline: Academics find eight vulnerabilities in Android's VoIP components

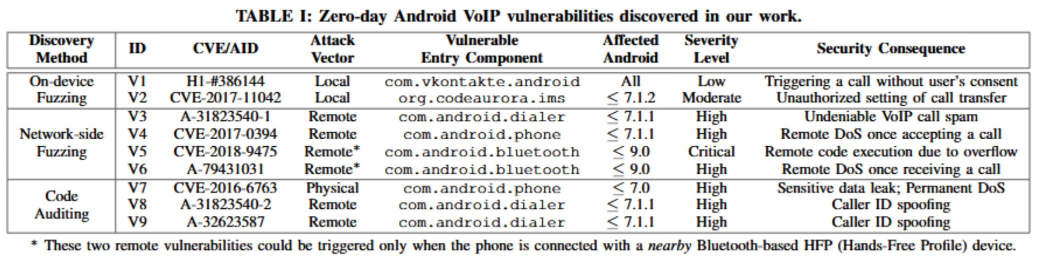


TABLE I: Zero-day Android VoIP vulnerabilities discovered in our work.

Discovery Method	ID	CVE/AID	Attack Vector	Vulnerable Entry Component	Affected Android	Severity Level	Security Consequence
On-device Fuzzing	V1	H1-#386144	Local	com.vkontakte.android	All	Low	Triggering a call without user's consent
	V2	CVE-2017-11042	Local	org.codeaurora.ims	$\leq$ 7.1.2	Moderate	Unauthorized setting of call transfer
Network-side Fuzzing	V3	A-31823540-1	Remote	com.android.dialer	$\leq$ 7.1.1	High	Undeniable VoIP call spam
	V4	CVE-2017-0394	Remote	com.android.phone	$\leq$ 7.1.1	High	Remote DoS once accepting a call
	V5	CVE-2018-9475	Remote*	com.android.bluetooth	$\leq$ 9.0	Critical	Remote code execution due to overflow
	V6	A-79431031	Remote*	com.android.bluetooth	$\leq$ 9.0	High	Remote DoS once receiving a call
Code Auditing	V7	CVE-2016-6763	Physical	com.android.phone	$\leq$ 7.0	High	Sensitive data leak; Permanent DoS
	V8	A-31823540-2	Remote	com.android.dialer	$\leq$ 7.1.1	High	Caller ID spoofing
	V9	A-32623587	Remote	com.android.dialer	$\leq$ 7.1.1	High	Caller ID spoofing

* These two remote vulnerabilities could be triggered only when the phone is connected with a *nearby* Bluetooth-based HFP (Hands-Free Profile) device.

Below is a breakdown of all the nine bugs:

V1: INITIATE A VOIP CALL IN THE VK APP
A bug in how the Android Intent API interacts with the official VK (vKontakte) app can allow a malicious app installed on the device to start a VoIP call via the VK app and eavesdrop on the phone owner's nearby conversations.

No user interaction is needed to exploit this bug, and while it requires local access, the bug is ideal to be integrated inside Android spyware, remote access trojans (RATs), and other malware strains.

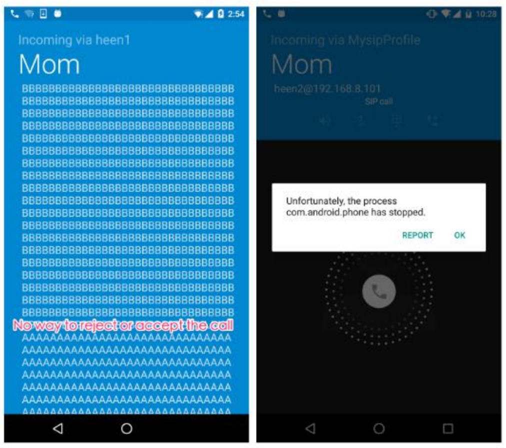
V2: UNAUTHORIZED CALL TRANSFER IN THE IMS INTERFACE
On-device malicious apps can misuse two local privileged APIs (in the QtiIMS VoIP component) to transfer incoming calls without authorization to an attacker's device

This was fixed back in 2017 when the issue was first discovered, and later received the CVE-2017-11042 identifier.

V3: UNDENIABLE VOIP CALL SPAM DUE TO LONG SIP NAME
This is the first of six remotely exploitable issues. According to researchers, attackers can initiate calls to a victim's phone using a long (1,043 characters) SIP name.

This SIP name fills up the user's phone screen and prevents them from answering or declining the call. If attackers chain multiple calls one after the other, they can prevent the user from using their phone during a critical period -- such as when performing another attack, such as hijacking an email account or bypassing 2FA.

"We call this kind of denial of service attack 'VoIP call bomb,' as similar to SMS bomb," researchers said. In current versions of the Android OS, Google limits the size of the SIP name in VoIP calls, to prevent abuse.


Incoming via heen1
Mom
No way to reject or accept the call

Incoming via MysipProfile
Mom
heen2@192.168.8.101
SIP call
Unfortunately, the process com.android.phone has stopped.
REPORT OK

Fig. 7: A demo of exploiting V3. Fig. 8: A demo of exploiting V4.

Publication: ZD Nets Online
Date: 01 October 2019
Headline: Academics find eight vulnerabilities in Android's VoIP components

V4: REMOTE DOS IN TELEPHONY ONCE ACCEPTING A CALL
The fourth bug can also be exploited remotely. Attackers can use malformed SDP packets to crash a victim's device when accepting an incoming call (see image above).

Just like the previous two bugs, this was fixed in 2017, after the researchers' report.

V5: REMOTE CODE EXECUTION DUE TO STACK BUFFER OVERFLOW
The worst bug researchers found was CVE-2018-9475, fixed in Android Oreo, in 2018. This is a remote code execution (RCE) issue that can allow attackers to run malicious code on a remote device via a VoIP call.

Researchers found that they could trigger a stack buffer overflow if a user name (or caller number) in a VoIP call had more than 513 bytes.

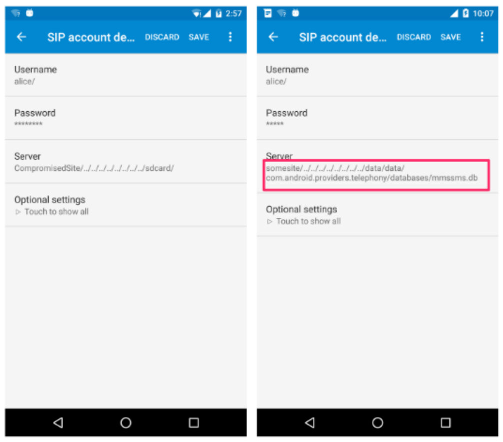
"This vulnerability allows an adversary to overwrite the return address of the ClccResponse function, causing remote code execution," researchers said. This bug affected all Android versions up to v9 (Pie), included.

V6: REMOTE DOS IN BLUETOOTH ONCE RECEIVING A CALL
V6 is similar to V5, and is also a bug that can also be exploited by making VoIP calls with long caller numbers, but this one only causes phones to crash, rather than allow attackers to run code on the device.

V7: DATA LEAK AND PERMANENT DOS DUE TO PATH TRAVERSAL
The seventh bug they found is not remotely exploitable, meaning it can only be abused by other malicious apps installed on the device.

However, this bug is pretty serious. It's a classic path traversal issue that happens because the SIP protocol and the Android treat the ".." and "/" characters differently. The researchers explain:

For example, by physically setting "sipuser" and "serverip" in the format of Fig. 10(a), mProfileDirectory becomes "/data/data/-com.android.phone/files/alice/@SomeSite/../../../../../../sdcard/" and leaks the sensitive SIP profile file to the public SD card. A permanent DoS could also happen if "serverip" is set to overwrite another system app's file, e.g., mmssms.db shown in Fig. 10(b).



Username
alice/

Password
****

Server
CompromisedSite/../../../../../../../../sdcard/

Optional settings
▷ Touch to show all

Username
alice/

Password
*****

Server
somesite/../../../../../../../../data/data/
com.android.providers.telephony/databases/mmssms.db

Optional settings
▷ Touch to show all

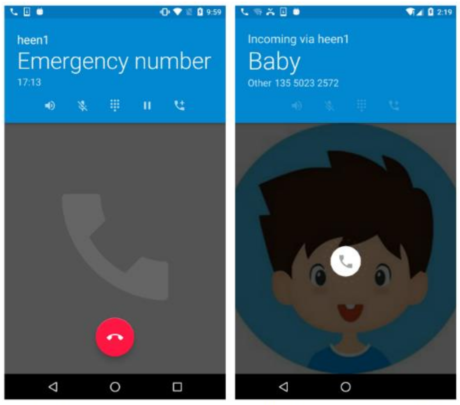
(a) Leaking data to SD card. (b) Causing permanent DoS.

Fig. 10: Demo screenshots of exploiting the vulnerability V7.

Publication: ZD Nets Online
Date: 01 October 2019
Headline: Academics find eight vulnerabilities in Android's VoIP components

V8: CALLER ID SPOOFING DUE TO MIS-PARSING "&"
This vulnerability is due the PSTN (Public Switched Telephone Network) number format, which does not account for the "&" character.

Incoming VoIP calls containing "&" characters result in the Android OS reading only the numbers before the character, but not after, allowing for easy caller ID spoofing.



(a) Spoofing as an emergent number. (b) Spoofing as a contact number.

Fig. 11: Demo screenshots of exploiting the vulnerability V8.

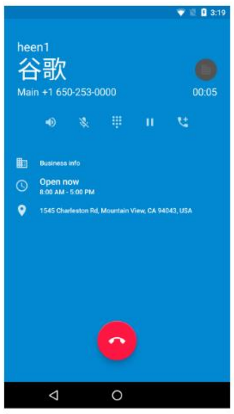


Fig. 12: A demo screenshot of exploiting the vulnerability V9.

V9: CALLER ID SPOOFING DUE TO "PHONE-CONTEXT"
The ninth and last vulnerability is also due to the PSTN (Public Switched Telephone Network) number format, but this time via the "phone-context" PSTN parameter. The research team explains.

For example, in PSTN's convention, the number "650253000;phone-context=+1" is equivalent to "+1650253000", where the value of "phone-context" becomes the prefix of the number. However, such convention should not apply to VoIP calls, which is unfortunately ignored by the dialer app. As a result, an adversary can intentionally set the caller number as "650253000;phone-context=+1", and the dialer app will interpret it as "+1650253000" and display it as Google's call, which is clearly presented in Fig. 12.
The full white paper is named "Understanding Android VoIP Security: A System-level Vulnerability Assessment" and is available for download here.

The paper's authors come from the OPPO ZIWU Cyber Security Lab in Shenzen, China; The Chinese University of Hong Kong; and the Singapore Management University.