

# Forum:[Hack Forums](#)

Thread: [Essay on Stage-fright](#)

**Nomad.Null** - September 19, 2015, 2:03 pm

Stage Fright Vulnerability: What does it mean and how to handle it

The latest vulnerability in the news, which has gotten the attention of hundreds of users of Android smartphones. The vulnerability claims to call into question the security of Android phones and is leaving a lot of users with concerns about whether or not to stick with the Google-owned and managed Smartphone Operating System.

There are quite a few issues surrounding this problem, but it is not necessarily as serious as many people claim and it is important to remember that the news likes to make things sound a lot more exciting, dangerous, and catastrophic than they may actually be. After doing some research and diving deep into the web and hacker-underground BBS forums, I have come to realize the true implications and usefulness of the Stage Fright Vulnerability and what it means for android users. My goal here is to explain and describe the implications of stage-fright and what users of Android smartphones can do to mitigate the threat.

What is stage-fright?

Stage-fright is a multimedia library with extremely high privileges that was built into the Android OS for the purpose of handling and parsing media files. What it does is, it will detect an incoming multimedia file such as a video over MMS, and it interprets the video encoding prior to viewing so that it is ready to be viewed by the user whenever they wish. We have all heard, thus-far, that stage-fright is vulnerable to MMS messages. While this is true, it is also vulnerable to drive-by downloads, embedded webpage media, and other vectors which an attacker can use to send a malicious file to your device.

What does the vulnerability do?

The problem: because of numerous oversights by developers, this library does not properly sanitize its functions, which can cause numerical integers to exceed their memory address space. In layman's terms, it is possible to force stage-fright to process more data at one time than it was designed to allow. This causes what is known as a buffer-overflow. If you are unfamiliar with this term, this is like filling up a cup with water until it overflows, spilling out into the system. Anything that spills out of the allocated memory space is then read and executed without regard to whether or not it is malicious.

That means that whatever is included in the video is allowed to execute without regard to what might be in it. Once this overflow occurs, the operating system does not check for malicious code, and, in fact, does not care if any code exists in it at all, nor does it throw any errors. This, in turn, allows the code to execute on the phone in the background, where there are no administrative restrictions.

In a worst case scenario, an expert of the Android OS, particularly a hacker, can use this to manipulate the system and cause it to execute whatever code they pre-package into the video. Because videos can be quite large, that means that viruses can be hid in the video that are designed to execute whether or not the user opens the video.

Who is affected by the Vulnerability?

Unfortunately, this vulnerability is wide-spread and it affects everyone using Andorid OS version 2.2 or higher. It was discovered to have been in place for the last several years; however, it should be noted that no single user that has ever reported having their phone hacked was a victim of this type of exploit.

How bad could it get?

Truly skilled hackers, particularly those malware-engineers out there who are familiar with the C and C++ languages will be able to run any amount of malicious code against a target. The package itself can be carried by any media file, or generated by a script such as Python. In a worst case scenario, the hacker can gain full access to all of the phone's files and system controls, seize information on the user such as banking and credit card information, contacts, pictures, videos, and documents.

How easy is it?

In theory, it's as simple as attaching a virus to a video and sending it to a random phone number, and boom you have full access.

In actuality, this is not an easy feat to accomplish. In my experience, the grand majority of Black-Hat (criminal) hackers, or Crackers out there in the world, are actually highly inexperienced or totally unfamiliar with coding. The C languages are particularly difficult to master, particularly when being accompanied or generated by other code. The code of the exploit itself is quite complex, as I have found, and it is difficult to ensure its stability and accuracy when trying to execute against a target. Much of the exploit relies on the attacker's familiarity with Android, as well as a great deal of intuition because they are required to make highly-educated guesses at the memory address spaces in order for their code to execute correctly. For an inexperienced hacker, who are called "script kiddies" or 'skids' are largely incapable of performing such a sophisticated attack.

Could I be infected?

Not likely... Google has already done preliminary scans and discovered no traces of this exploit actually being taken advantage of \*yet\*. This simply means that it was not discovered until recently, which we know to be true. This also means that in order for someone to truly take advantage of this, they would have to re-create the exploit and discover how to take advantage of it from scratch, which took the original penetration tester quite a large amount of dedicated time to discover.

The added fact that seasoned hackers would need to do this from scratch, that the average and new hackers on the scene (which number in the hundreds of thousands...) could never pull this off without help.

In reality, most hackers will keep this exploit code to themselves, due to its high profitability and difficulty to develop. This is not something that would be handed out lightly. For a hacker to create an exploit kit to generate this code with just a few button clicks, as most script-kiddies prefer, would take a tremendous amount of work, and I would compare it to handing a kid a grenade. Despite all of the danger surrounding stage-fright, the threats can still be mitigated.

What can I do about it?

Simple. Install Anti-virus for your phone. ESET (which I highly recommend), Avast!, Norton, Avira,

AVG, and other known antivirus providers have developed mobile, Android-versions of their antivirus software that are already familiar with and can easily stop this type of attack. Buffer overflows are not an uncommon exploit, and are easily identified by any of these antiviruses. These experts of IT Security are no strangers to viruses, and in fact, are totally prepared for this scenario. By installing Anti-virus, your chances of actually falling victim to this attack will be significantly reduced.

Not to mention the fact that someone with extremely in-depth knowledge about Android OS and hacking combined would have to specifically target you in order to make this happen.

Could it happen? Absolutely. Will it happen to you? Probably not. And that's good news for us Android users.

### Final Words

This has not compromised my faith in Android OS's security. As a penetration tester, I'm already more than aware that there is no guarantees when it comes to security or technology. But I do know that you can certainly improve your chances by being cautious, conscious, and prepared for these kinds of events. Simply install an Anti-virus, and your chances of losing the battle against the hackers has been significantly reduced. As always, you should avoid shady websites, untrustworthy products, and recklessly downloading anything you find on the internet with your phone.

I would encourage everyone to maintain vigilance and to maintain the best possible security practices as they always have. As long as users continue to educate themselves about mobile security and ensure they have installed and implemented a trust antivirus program, the likelihood of actually falling victim to this type of attack is slim-to-none.

Educate yourselves and prepare for the worst, but don't panic. The world isn't coming to an end. Android still lives.

[Edited for corrections: 9/19/2015 per xorl's input.]

Ever your faithful servant, - Your friendly-neighborhood Demon ?

**PuntoSinclair** - September 19, 2015, 2:20 pm

Hi, you sent me a PM, I can't reply you because I don't have 25 posts yet, if you can send me your contact would be perfect. thanks

**xorl** - September 19, 2015, 5:55 pm

You have no idea what you're talking about.

Quote:Stage-fright is a background process that was built into the Android OS for the purpose of making video-streaming faster and more convenient for its users. Stagefright is a multimedia library for audio and video handling.

Quote:What it does is, it detects when video has been sent over a MMS (Multimedia Message Service) message, and it preloads the encoded video so that it is ready to view whenever the user wishes to look at it. You're way off here. While autoretrieving MMS is the "dangerous" part about it, stagefright is used for most media handling in Android. So MMS (it's being used even if you disable preloads), browsers, file downloads, emails, and a bunch of other stuff are using the stagefright library.

To quote jduck: "If an Android device is allowing you to view media from any untrusted person then it's probably affected".

Quote:The problem: because of an oversight by developers, this service does not properly sanitize the video encoding to search for malicious and arbitrary code. That means that whatever is included in the video is allowed to execute without regard to what might be in it. It does not check for malicious code, and, in fact, does not care if any code exists in it at all, nor does it throw any errors. This, in turn, allows the code to execute on the phone in the background system, what is called the "kernel." What the fuck are you on about? We're talking about memory corruption vulnerabilities here. It's not going to interpret any code you throw at it and just execute it...

It's several integer overflows that can be turned into heap overflows. This can then be exploited into running arbitrary code.

Quote:Truly skilled hackers, particularly those malware-engineers out there who are familiar with languages such as RUBY, Python, and Android's script (based on Linux BASH) will be able to run any amount of malicious code against a target. fucking lol again, this shows just how much you've "researched" the topic before making your thread. Please show us ANY payload that consisting of bash, ruby or python which get executed by stagefright.

Your entire thread is based on misinformation, false claims and a lacking technological understanding of the topic.

For anyone actually interested in understanding stagefright:

[blog.zimperium.com/stagefright-vulnerability-details-stagefright-detector-tool-released/](http://blog.zimperium.com/stagefright-vulnerability-details-stagefright-detector-tool-released/)

[blackhat.com/docs/us-15/materials/us-15-Drake-Stagefright-Scary-Code-In-The-Heart-Of-Android.p  
df](http://blackhat.com/docs/us-15/materials/us-15-Drake-Stagefright-Scary-Code-In-The-Heart-Of-Android.pdf)

[youtube.com/watch?v=71YP65UANP0](http://youtube.com/watch?v=71YP65UANP0)

[googleprojectzero.blogspot.ca/2015/09/stagefrightened.html](http://googleprojectzero.blogspot.ca/2015/09/stagefrightened.html)

[blog.exodusintel.com/2015/08/13/stagefright-mission-accomplished/](http://blog.exodusintel.com/2015/08/13/stagefright-mission-accomplished/)

**Nomad.Null** - September 19, 2015, 6:30 pm

I was wondering how long it would take you go show up again, Ninjury.

I'm not going to even waste my breath on you. Nothing I said was inaccurate. You can bitch about the technicalities and generalities all you want.

User has 7 matching IPs, and is responsible for thousands of dollars worth of scams. Don't trust.

I withdraw this accusation.

**xorl** - September 19, 2015, 6:34 pm

(Sep 19, 2015 06:30 PM)True Demon Wrote: [\[CITE\]](#) I was wondering how long it would take you go show up again, Ninjury.

I'm not going to even waste my breath on you. Nothing I said was inaccurate. You can bitch about the technicalities and generalities all you want.

User has 7 matching IPs, and is responsible for thousands of dollars worth of scams. Don't trust.

I have no idea what you're talking about, I'm not the person you think I am.

Everything I said is true and if you think otherwise you should point out where I'm wrong.

**Nomad.Null** - September 19, 2015, 7:27 pm

(Sep 19, 2015 05:55 PM)xorl Wrote: [\[CITE\]](#) You have no idea what you're talking about.

Stagefright is a multimedia library for audio and video handling.

You are correct. I mispoke. However, the fact remains that the multimedia library handles numerous background processes that read any given media that is pulled into the device. Seeing as it is called by dozens of background applications, I can easily see where I made my mistake.

xorl Wrote: You're way off here. While autoretrieving MMS is the "dangerous" part about it, stagefright is used for most media handling in Android. So MMS (it's being used even if you disable preloads), browsers, file downloads, emails, and a bunch of other stuff are using the stagefright library.

To quote jduck: "If an Android device is allowing you to view media from any untrusted person then it's probably affected".

After reviewing Drake's presentation, I admit you are right here as well. I neglected to notice that stage-fright even handles something as seemingly innocent as the metadata for thumbnail images...something that can easily be altered for executing an overflow.

xorl Wrote: What the fuck are you on about? We're talking about memory corruption vulnerabilities here. It's not going to interpret any code you throw at it and just execute it...

It's several integer overflows that can be turned into heap overflows. This can then be exploited into running arbitrary code.

My statement was generalized, but not incorrect here. The lack of sanitization of the code, like any vulnerability, is the problem here, which leads to the execution of arbitrary code. I wrote this essay in such a way as it could be understood by the layman in order to explain the concept and function of the stage-fright vulnerability. As you said, it is exploited into executing arbitrary code. I was not trying to get overly technical. This whole essay was about explaining the actual danger and how practical this vulnerability is and to alleviate the fear it has caused. At the heart of it all, this is still a vulnerability that takes advantage of a buffer overflow, something that any decent Anti-virus is able to detect and stop, which was the point I was trying to make in the end. It is an extreme vulnerability, no doubt, but with so many people panicking, thinking that they're going to have their bank-accounts emptied on any given day...I thought I should clear a few things up about how likely that would be. In no uncertain terms, I can say with confidence, you have a better chance of getting struck by lightning.

xorl Wrote: fucking lol again, this shows just how much you've "researched" the topic before making your thread. Please show us ANY payload that consisting of bash, ruby or python which get executed by stagefright.

The original proof of concept consisted of a python script which generated an MP4 file, according to several articles I had discovered. I did, however, neglect to notice that the actual payload contained malicious C. I admit to being in the wrong here as well. I misread the article, believing that the entire exploit was written in Python. It was then my assumption that any further arbitrary code would have then been executed in Python, Ruby or run by a bash function. As they say, "To assume makes an ass out of 'u' and me."

However, my implication regarding Bash and Ruby was not for the payload itself, but rather for manipulating files and running malicious code that would otherwise damage, alter, or transmit its files. Clearly, I should have clarified this.

Again, I was not actually investigating the source of any exploit-code. I was merely explaining and elaborating on its concept, utility, and likelihood of being targetted against the average user. Simply put, I did not find it necessary to get technical here, given the purpose of this essay.

I withdraw my accusations and will ammend this thread appropriately to reflect your corrections. I'd appreciate it if, in the future, you'd be a little bit more civil about your corrections and criticism. I'm not afraid to admit to being wrong about something, and I would have edited this immediately to

reflect that, had you been more courteous instead of flame-baiting me.

For the record, Ninjury is a 'reknowned' scammer that has gone by many names on this forum, and often lashed out at me and other members like me much in the same way you did. In particular, he has targeted me several times, hence why I suspected you and he were one and the same. I still maintain that your 7 matching IP addresses is highly suspicious. Nevertheless, I will withdraw the down-rep and accusation.

**lolx45** - September 19, 2015, 9:00 pm

(Sep 19, 2015 07:27 PM)True Demon Wrote: [\[CITE\]](#) At the heart of it all, this is still a vulnerability that takes advantage of a buffer overflow, something that any decent Anti-virus is able to detect and stop, which was the point I was trying to make in the end. It is an extreme vulnerability, no doubt, but with so many people panicking, thinking that they're going to have their bank-accounts emptied on any given day...I thought I should clear a few things up about how likely that would be. In no uncertain terms, I can say with confidence, you have a better chance of getting struck by lightning.

1. I don't think you understand the concept of antiviruses.
2. "you have a better chance of getting struck by lightning" - ever heard of mass spamming? You don't need to be targeted individually to get compromised.

**Nomad.Null** - September 19, 2015, 11:16 pm

(Sep 19, 2015 09:00 PM)lolx45 Wrote: [\[CITE\]](#) 1. I don't think you understand the concept of antiviruses.

2. "you have a better chance of getting struck by lightning" - ever heard of mass spamming? You don't need to be targeted individually to get compromised.
- 1: The 'concept' of antiviruses? There's nothing conceptual about what I've said. It's anomaly-based detection. It's a common virus-detection method, and it is the most successful in stopping buffer overflows cold, particularly zero-days. Signature-based detection isn't worth a damn against stage-fright right now, or any zero-day for that matter.

- 2: Mass spamming a video with a malicious payload isn't exactly efficient...or quiet. Entirely feasible, but it's an easy way to get caught. You're talking about a several megabyte payload as opposed to a few kilobytes...