15

[logkitty] RCE via insecure command formatting

Share:

TIMELINE

ik317 submitted a report to Node.js third-party modules. Mar 20th (3 years ago)

I would like to report a `RCE` issue in the `logkitty` module.
It allows to execute `arbitrary commands remotely inside the victim's PC`

Module

module name: `logkitty`
version: `0.7.0`
npm page: `https://www.npmjs.com/package/logkitty`

Module Description

> Display pretty Android and iOS logs without Android Studio or Console.app, with intuitive Command Line Interface.

Module Stats

[170,222] downloads in the last week

Vulnerability Description

The issue occurs because a `user input` is formatted inside a `command` that will be executed without any check. The issue arises here:
https://github.com/zamotany/logkitty/blob/master/src/android/adb.ts#L55

Steps To Reproduce:

1. Check there aren't files called `HACKED`
2. Execute the following commands in another terminal:

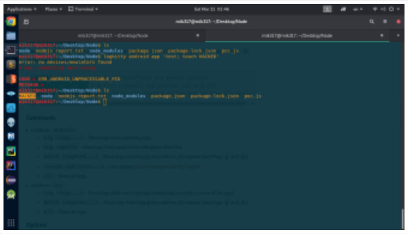
Code 198 Bytes Wrap lines Copy Download

```
npm i logkitty # Install affected module
logkitty android app 'test; touch HACKED' #  Note the *touch command* is inside the *'* (single quote), so it's an argument, while it will be executed anywa
```

1. Recheck the files: now `HACKED` has been created :)

Image F754955: Screenshot_from_2020-03-21_01-46-19.png 179.84 KiB

Zoom in Zoom out Copy Download



Patch

> Don't format `commands` using insecure `user's inputs` :)

Supporting Material/References:

- [OPERATING SYSTEM VERSION]: Kali Linux
-
-

Wrap up

- I contacted the maintainer to let them know: [N]
- I opened an issue in the related repository: [N]

Impact

`RCE` via command formatting on `logkitty`

1 attachment:
F754955: Screenshot_from_2020-03-21_01-46-19.png

nochnoidozor posted a comment. Mar 21st (3 years ago)

Hi @mik317,

Thank you for your submission. Your report is currently being reviewed and the HackerOne triage team will get back to you once there is additional information to share.

nochnoidozor changed the status to ● Triaged. Mar 21st (3 years ago)

Hello @mik317,

Thank you for your submission! We were able to validate your report, and have submitted it to the appropriate remediation team for review. They will let us know the final ruling on this report, and when/if a fix will be implemented. Please note that the status and severity are subject to change.

Regards,
@nochnoidozor

zamotany joined this report as a participant. Apr 7th (3 years ago)

zamotany posted a comment. Apr 7th (3 years ago)

With the following PR: https://github.com/zamotany/logkitty/pull/18, reproduction steps are no longer producing `HACKED` file. The PR was released to NPM as `0.7.1`.

marcinhoppe Node.js third-party modules staff posted a comment. Apr 7th (3 years ago)

@zamotany many thanks for getting back to us so quickly!

@mik317 @nochnoidozor can you verify that version `0.7.1` properly fixes the vulnerability?

nochnoidozor posted a comment. Apr 7th (3 years ago)

Hi @mik317 - Can you help us verify if the issue has been properly fixed in version `0.7.1`?

Thanks for your help!
@nochnoidozor

ik317 posted a comment. Updated Apr 7th (3 years ago)

Hi @zamotany :),
thank you for the fast response and fix :).
I can confirm the fix patches the `command injection` issue is fixed properly :)

Thank again for the hard work :)
Best regards,
Mik

marcinhoppe Node.js third-party modules staff posted a comment. Apr 8th (3 years ago)

@mik317 thanks for quick verification!

@zamotany is it OK for you that we disclose this vulnerability now?

zamotany posted a comment. Apr 8th (3 years ago)

Sure

marcinhoppe Node.js third-party modules staff closed the report and changed the status to ● **Resolved**. Apr 9th (3 years ago)

marcinhoppe Node.js third-party modules staff updated the severity from Critical to High (7.8). Apr 9th (3 years ago)

marcinhoppe Node.js third-party modules staff requested to disclose this report. Apr 9th (3 years ago)

marcinhoppe Node.js third-party modules staff posted a comment. Apr 9th (3 years ago)

@mik317 i adjusted the CVSS score for this vulnerability. I classified this as a local attack, because it seems unlikely that this package would be used remotely. Let me know if this score makes sense to you.

ik317 posted a comment. Apr 9th (3 years ago)

Hi @marcinhoppe :)
Yeah, I agree the attack could be local, anyway it depends mostly on the implementation (someone could have used the package as part of a server which could allow someone analyzing an APK remotely, and that would have lead to `remote` attack vector).

Anyway, if it's fair for you, it's ok also for me ;)

Thank you again,
Mik

marcinhoppe Node.js third-party modules staff posted a comment. Apr 14th (3 years ago)

I requested a CVE for this finding: CVE-2020-8149.

ik317 posted a comment. Updated Apr 14th (3 years ago)

Hi @marcinhoppe :),
thank you so much for this :). It's simply amazing :).
Hope to work with you again in the near future, and most important that's going all well by your side of the screen ;). Stay safe :).

Thank again and regards, Mik

Hi @nochnoidozor :),
can we pls update the CVE reference of this report with the CVE provided by @marcinhoppe ?

Best, Mik