New issue

Jump to bottom

RVD#3316: No authentication in MAVLink protocol #3316

Open vmayoral opened this issue on Jun 30, 2020 · 22 comments

Labels components software robot component: Ardupilot robot component: MAVLink robot component: PX4 severity: critical version: 1.0 vulnerability

vmayoral commented on Jun 30, 2020 • edited Member

```yaml
id: 3316
title: 'RVD#3316: No authentication in MAVLink protocol'
type: vulnerability
description:  The Micro Air Vehicle Link (MAVLink) protocol presents no authentication
  mechanism on its version 1.0 (nor authorization) whichs leads to a variety of attacks
  including identity spoofing, unauthorized access, PITM attacks and more.
  According to literature, version 2.0 optionally allows for package signing which
  mitigates this flaw. Another source mentions that MAVLink 2.0  only provides a simple
  authentication system based on HMAC. This implies that the flying system overall
  should add the same symmetric key into all devices of network. If not the case, this may
  cause a security issue, that if one of the devices and its symmetric key are compromised,
  the whole authentication system is not reliable.
cwe: CWE-306
cve: CVE-2020-10282
keywords:
- MAVLink
- v1.0
- v2.0
- PX4
- Ardupilot
system: "MAVLink: v1.0"
vendor: "PX4"
severity:
  rvss-score: 9.6
  rvss-vector: RVSS:1.0/AV:AN/AC:L/PR:N/UI:N/S:U/Y:T/C:H/I:H/A:H/H:U
  severity-description: critical
  cvss-score: 9.8
  cvss-vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
links:
- https://arxiv.org/abs/1906.10641
- https://arxiv.org/abs/1905.00265
- https://ieeexplore.ieee.org/document/8425627
- https://www.researchgate.net/publication/335973981_Assessing_and_Exploiting_Security_Vulnerabilities_of_Unmanned_Aerial_Vehicles
- https://link.springer.com/chapter/10.1007/978-981-13-8406-6_66
- https://www.esat.kuleuven.be/cosic/publications/article-2667.pdf
- https://www.usenix.org/conference/usenixsecurity19/presentation/kim
- https://docs.google.com/document/d/1ETle6qQRcaNWAmpG2wz0oOpFKSF_bcTmYMQvtTGI8ns/edit
- https://docs.google.com/document/d/1upZ_KnEgK3Hk1j0DfSH19AdKFMoSqkAQVeK8LsngvEU/edit
- https://docs.google.com/document/d/1XtbD0ORNkhZ8eKrsbSIZNLyg9sFRXMXbsR2mp37KbIg/edit
- https://github.com/PX4/Firmware/issues/13538#issuecomment-574281772
- https://github.com/rligocki/Diploma_thesis_px4
flaw:
  phase: unknown
  specificity: subject-specific
  architectural-location: platform code
  application: Flying vehicles and/or others using MAVLink protocol.
  subsystem: communication
  package: N/A
  languages: C, C++
  date-detected:
  detected-by:
  detected-by-method: testing
  date-reported: '2020-06-30'
  reported-by: "Victor Mayoral Vilches (Alias Robotics)"
  reported-by-relationship: security researcher
  issue: https://github.com/aliasrobotics/RVD/issues/3316
  reproducibility: always
  trace: N/A
  reproduction: N/A
  reproduction-image: N/A
exploitation:
  description: Not available
  exploitation-image: Not available
  exploitation-vector: Not available
  exploitation-recipe: ''
mitigation:
  description: MAVLink 2.0 includes signing capabilities which mitigate this issue. Signatures seem to be optional for backwards compatibility and https://arxiv.org/abs/1906.10641 con
  pull-request: N/A
  date-mitigation: null
```

vmayoral added components software vulnerability robot component: PX4 robot component: Ardupilot robot component: MAVLink version: 1.0 severity: critical labels on Jun 30, 2020

vmayoral commented on Jun 30, 2020 Member Author

Been reviewing the release notes of PX4 and from the outlook, versions prior than `v1.7.0` (stable) don't seem to fully support all the features including with the signing capabilities. A similar research should be done with Ardupilot and other autopilots that make use of this protocol.

NOTE: for the sake of time, no actual code review was done to confirm the presence and/or implementation of MAVLink 2.0 in `v1.7.0` of PX4 so take the information above only as preliminary

For reference: https://github.com/PX4/Firmware/releases?after=v1.7.4beta

vmayoral commented on Jul 1, 2020 Member Author

Further extended ticket with:

> MAVLink protocol provides a very good way to transport a huge amount of data with very low overhead but provides only very basic security features. In current state MAVLink only provides a simple authentication system based on HMAC. This means that drone manager must add the same symmetric key into all devices of net- work. This creates a security issue, that if one of the devices and its symmetric key are compromised, the whole authentication system is not reliable. Then messages might be suppositious and there is no way to find it out.

From PX4/PX4-Autopilot#13538 (comment)

glerapic commented on Jul 1, 2020

This has the potential to harm humans if a bad operator takes control of the device and it goes kamikaze. Please consider changing the Hazard value from H:U to H:H

vmayoral commented on Jul 1, 2020 Member Author

Mmmm I'm a bit hesitant about that. This vuln itself affects the robot components and would need to be chained with other flaws to affect humans, don't you think so? That's why i left it as *unknown*.

glerapic commented on Jul 1, 2020

Fair enough.

LGTM!

vmayoral commented on Jul 3, 2020 Member Author

Fine then, filed for a CVE ID CVEProject/cvelist#4246

khancyr commented on Aug 7, 2020

Hello,

I don't understand the point of this ticket... It is clearly stated that mavlink 1.0 protocol doesn't provide any system of authentification....
So this vulnerability isn't one...

About mavlink2.0 with signing, if somebody got the signing key, yes you got an issue but that like for all systems when you lose a key.

About ArduPilot, we use mavlink2.0 by default since some time. So we aren't vulnerable to this.

vmayoral commented on Aug 8, 2020 Member Author

Hi **@khancyr**,

> I don't understand the point of this ticket... It is clearly stated that mavlink 1.0 protocol doesn't provide any system of authentification....

That's right. MAVLink 1.0 does not implement authentication or authorization but while that design decision is respectable (and likely appropriate for that time), it still includes with it a security flaw which is what's described in this ticket and what's captured with a CVE ID.

The lack of authentication is thereby a vulnerability on MAVLink's version 1.0.

> About mavlink2.0 with signing, if somebody got the signing key, yes you got an issue but that like for all systems when you lose a key.
>
> About ArduPilot, we use mavlink2.0 by default since some time. So we aren't vulnerable to this.

Last time I checked the source code of PX4 and ArduPilot (a couple of months ago) noticed that for backwards compatibility (and to support several GCSs), the autopilot allows to negotiate communications with both MAVLink 1.0 and 2.0 endpoints. Depending on how intialization and handshake process is established, it's somewhat trivial to get the autopilot to exchange messages using the insecure and unauthenticated version 1.0. This is somewhat covered at https://mavlink.io/en/guide/mavlink_version.html#negotiating_versions.

I cooked a simple exploit PoC for PX4 which validated this. Never got the time or resources to test it also in ArduPilot but I'm happy to review it if anyone can support us in the process with some resources.

Frankly, I scrolled quickly through the ArduPilot code so I might have missed something but happy to double check it if you can provide pointers where there's evidence that ArduPilot implements only a subset of MAVLink, and thereby discards this characteristic (negotiation of the version). That certainly would be a proper mitigation, though it'll break backwards compatibility.

Let me now if you something to add in here but for the time being, I'm keeping this open based on this rationale for ArduPilot.

 **vmayoral** mentioned this issue on Aug 8, 2020

RVD#3317: MAVLink version handshaking allows for an attacker to bypass authentication #3317

Open

khancyr commented on Aug 8, 2020

That is only an issue if you are using mavlink protocol with SiK or RFD like radio (even if this have aes encryption) because they are serial over the air radio and thus implement OSI layer 3 and 4 with MAVLINK.

On other case, like with IP protocola, MAVLINK isn't subject to vunerability as it implements just the layer 7, mavlink is just a binarised messaging system such as protobuf or rosmsg.

So that insn't really the MAVLINK protocol that is vunerable but the way to distribute mavlink messages.

Second point, on the drone PX4 and ArduPilot are using MAVLINK on multiple serial as layer 3 and 4 without authentication but I doubt that is an issue as that is just a way to pass message between sensor, such as I2C/SPI/1-wire etc. Is that a vunerability to have unsecured communication between sensors on a drone ?

In conclusion, where is the issue with MAVLINK? On the radio using it : sure. On the protocol itself ? I fail to see where...

Ps: a proper analysis of mavlink system would have shown you that some implementations are subject to heartbleed like vulnerability...

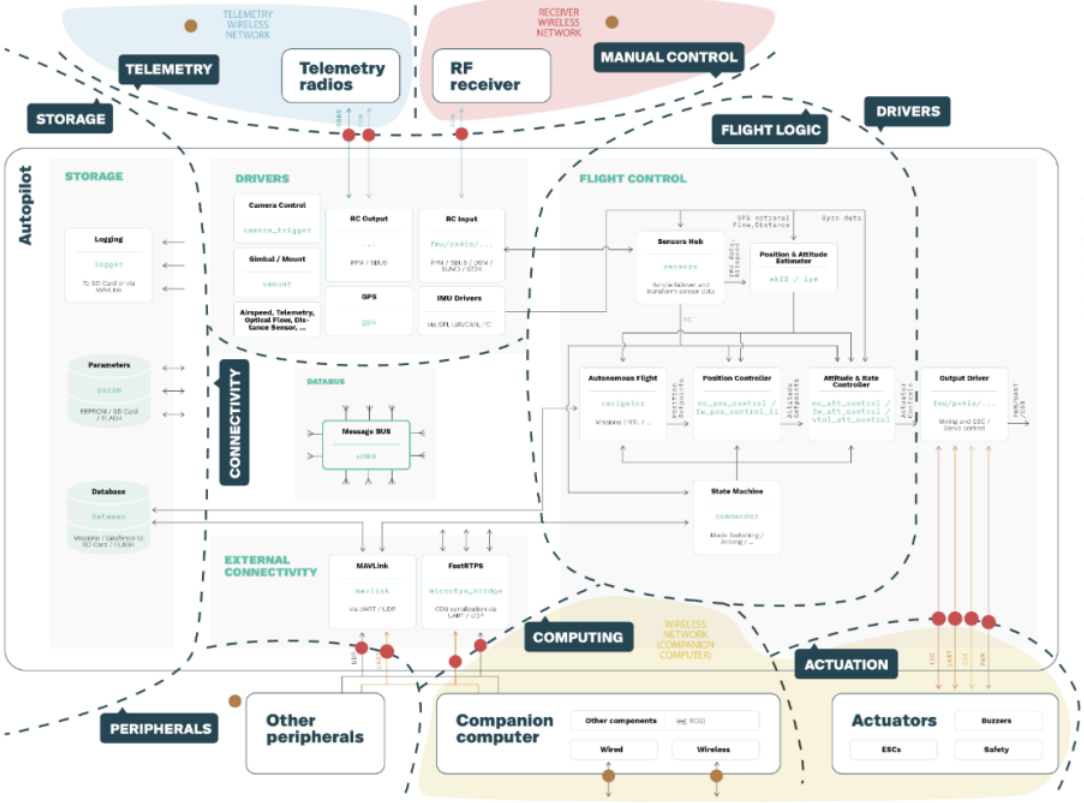
vmayoral commented on Aug 8, 2020 Member Author

> That is only an issue if you are using mavlink protocol with SiK or RFD like radio (even if this have aes encryption) because they are serial over the air radio and thus implement OSI layer 3 and 4 with MAVLINK.

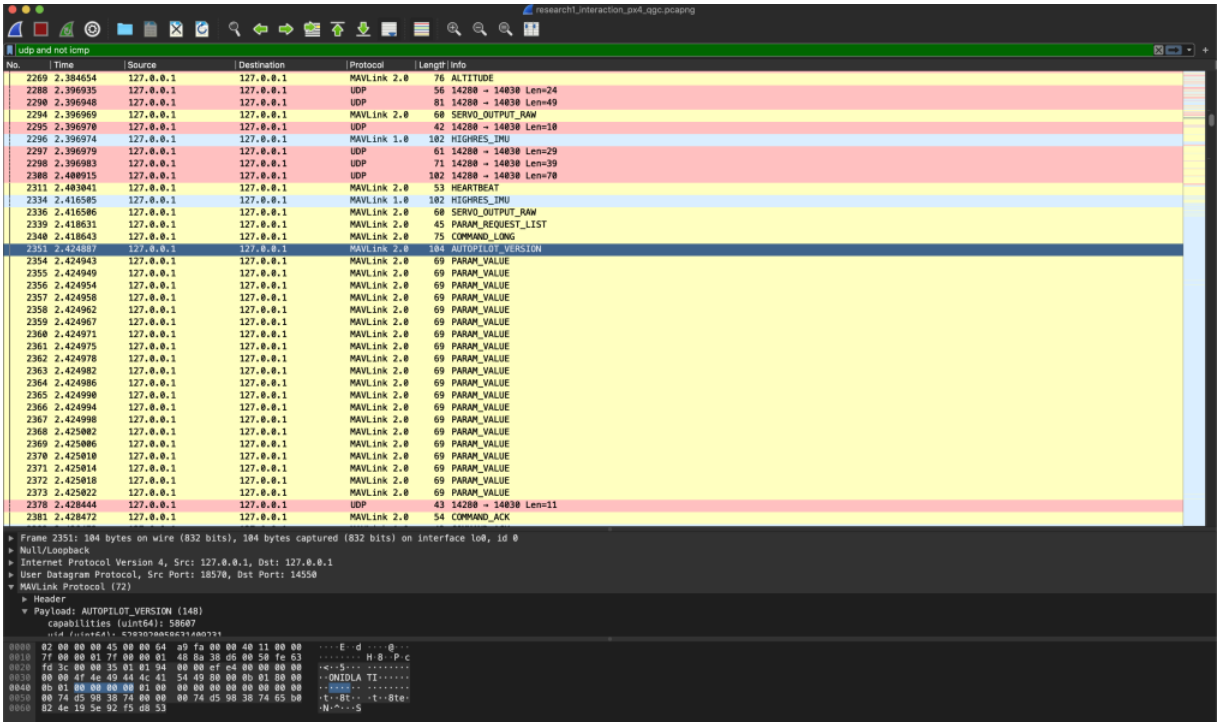
I agree that serial telemetry radios are subject to attack vectors exploiting this vulnerability.

> On other case, like with IP protocola, MAVLINK isn't subject to vunerability as it implements just the layer 7, mavlink is just a binarised messaging system such as protobuf or rosmsg.

I disagree kindly with this. My analysis focused not on ArduPilot but it's applicable to autopilots implementing MAVLink protocol and using it over any of its different (enabled) channels. The following graph from a recent presentation I gave at PX4 conference slides depicts a few examples:



Moreover, here's a screenshot from the analysis I performed recently of MAVLink as payload over UDP:



The whole protocol is implemented as a transport-layer payload. I dissected the protocol and crafted a few funny packages for the PoC demonstrating a few interesting issues such as the one discussed above. It's thereby possible over UDP/IP to act as a "malicious end-point/malicious GCS" and downgrade to MAVLink 1.0 in the "version negotiation" process, bypassing some of the security features enabled in MAVLink version 2.0.

Nevertheless, all this is to my understanding, is beyond the scope of this ticket and would require a somewhat knowledgeable attacker. Also, the ticket in here is just pointing to `version 1.0` (see labels of this ticket) which I believe we both agree are subject to what's being described. Let me know if we still have a disagreement and what actions would you suggest to take.

> Ps: a proper analysis of mavlink system would have shown you that some implementations are subject to heartbleed like vulnerability...

Interesting! Do you have a PoC or cycles to demonstrate something like this? I'd be more than happy to review a ticket coming from you and help/support collecting evidence.

vmayoral commented on Aug 11, 2020 — Member, Author

Hey **@khancyr**, do you have any additional input or insights on this?
Thanks,

khancyr commented on Aug 21, 2020

If you are using UDP then the transport layer is UDP not mavlink ...
As I said, if you want to use a some encryption you are free to do it to secure your mavlink message. We don't have encryption over USB too, but if you want to put some you can... that is the same for mavlink, just use a radio that have encryption and you will be fine, otherwise, yes everybody got access to you. That isn't a flaw nor vulnerability, that is in the specification, it was done expressively like that.

On the schema you gave MAVLINK comes from UDP or uart from another peripherical. UDP is securable easily, for uart you can just put some encryption on both side... That is what exist on RFD like radio. You have encrypted paired radio using mavlink...

OlivierJB commented on Nov 22, 2020 • edited

> that's right. MAVLink 1.0 does not implement authentication or authorization but while that design decision is respectable (and likely appropriate for that time), it still includes with it a security flaw which is what's described in this ticket and what's captured with a CVE ID.

This cannot be a CVE, and it is a misuse of the term. A CVE by definition requires that there is an underlying security policy, for which a vulnerability has been found. But there is no such policy, Or to put it differently that there was a design distinction between authorized, and unauthorized, access. Again not the case with Mavlink 1.0.

In short, this is like saying that a public place that has been made accessible to anyone has a CVE because it has a gate with no lock, or no fence.

Well ... no.

You could call Mavlink 1.0 insecure, or open. But that's about it.

vmayoral commented on Nov 24, 2020 • edited — Member, Author

Thanks **@OlivierJB** for jumping in and adding your view.

> This cannot be a CVE, and it is a misuse of the term. A CVE by definition requires that there is an underlying security policy, for which a vulnerability has been found.

Interesting. What you say makes sense since AFAIK MAVLink didn't consider security at its inception but at the same time, to my understanding (and research so far), a `CVE ID` by definition is a an entry in the `CVE list` containing an identification number (the ID), a description, and at least one public reference for a given publicly known cybersecurity vulnerability. I don't recall having seen that it's mandatory to identify a security policy behind the technology to consider filing IDs.

On what basis do you claim that a CVE can "only be assigned if there is an underlying security policy"? Can you provide a reference that help me dive deeper?

While it might be obvious to an experienced drone developer that MAVLink isn't secure (and on this basis you may judge, it's not necessary to have an ID), my current standpoint is that lots of technologies in robotics don't have such policies (security was completely disregarded) yet there're plenty of security vulnerabilities which should be registered and considered when designing a robotic application. An ID helps manufacturers (and other stakeholders) identify quickly what security issues affect a particular technology, regardless of their expertise with a flight stack or any of its underlaying technologies.

khancyr commented on Nov 24, 2020 • edited

then please open a CVE for :

- Wifi
- Bluetooth
- TCP
- UPD
- TCP/IP socket
- UART
- CAN
- I2C
- SPI
- 3G/4G/5G
- Lora
- XBEE
- Zigbee
- protobuf
- ZeroMQ
- MQTT
- gRPC
- ModBus
- PWM
- RC PWM
- PPM
- Oneshot
- CRSF
- DSM

- FPort
- IBus
- SBus
- SRXL
- ST24
- SUMD
- RS485
- UWB
- 1Wire
- FTP

Those are most of the protocol supported by ArduPilot and all are vulnerable

engranaabubakar commented on Nov 3, 2021

@khancyr I have read about MAVlink protocol which works with UDP in wireless/wifi. If we use WPA2 security at both ends ground control station and drone, then every MAVlink packet is encrypted with end to end encryption. How does someone sniffs or capture the drone? I am also confused about where the MAVlink protocol works in the UART scenario on the data link layer? Where is routing because it has an id of sender and destination id in numbers etc.?

engranaabubakar commented on Nov 3, 2021

@khancyr, as per your above comments, we need encryption at both ends by using UART, so these encryption standards should follow the MAVlink protocol's working requirements, and these algorithms will be vulnerable.

khancyr commented on Nov 3, 2021

Just encode your mavlink message, then crypt it with the algorithm you want and send it on UART... This has nothing to do with mavlink protocol

engranaabubakar commented on Nov 3, 2021

So no need to add authentication in the MAVlink protocol. Add another layer of security encryption on UART. I have a question can it affect the performance of drones like computational cost or latency after implementing any encryption algorithm on UART?

khancyr commented on Nov 3, 2021

Yes, encryption always come with some CPU usage

engranaabubakar commented on Nov 4, 2021

But the internet of drones is also a concern of MAVlink usages, if commands/Mavlink messages are traveling without encryption in-network, any adversary can capture these packets and use them for malicious purposes.

khancyr commented on Nov 4, 2021

such as any non-encrypted communication. If you want encryption, just use encryption on your transport layer.

if you chose to send raw tcp socket with mavlink message over the net, the issue isn't on mavlink but on your way to communicate. That apply to any communication

Assignees

No one assigned

Labels

components software robot component: Ardupilot robot component: MAVLink robot component: PX4 severity: critical version: 1.0 vulnerability

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

5 participants