IEEE P802.11  
Wireless LANs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Protecting against A-MSDU Attacks in Mesh Networks | | | | | |
| Date: 2025-10-01 | | | | | |
| Author(s): | | | | | |
| Name | Affiliation | Address | Phone | email |
| Mathy Vanhoef | KU Leuven | Celestijnenlaan 200a, 3000 Heverlee, Belgium |  | Mathy.Vanhoef@kuleuven.be |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Abstract

This submission proposes a defense to detect when a malicious outsider changes the A-MSDU Present subfield to 1 inside a mesh BSS (MBSS). This mitigates certain attacks in case signaling and payload protected A-MSDUs (SPP A-MSDUs) are not being used.

NOTE – Set the Track Changes Viewing Option in the MS Word to “All Markup” to clearly see the proposed text edits.

**Revision History:**

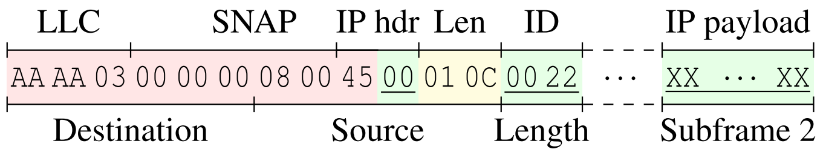
R0: Initial version.

R1: Added ‘discussion of the defense’ section.

**Discussion of the issue**

After the disclosure of the “FragAttacks” vulnerabilities, in particular CVE-2020-24588, an update was approved to detect if a malicious outsider turned an MSDU into an A-MSDU by changing the unauthenticated A-MSDU Present subfield in the QoS Control field to 1.

When an MSDU is turned into an A-MSDU in a nonmesh BSS, this can be detected by comparing the destination address of the first A-MSDU subframe to AA:AA:03:00:00:00. The following figure illustrates this check, where the bytes shown are those of an example MSDU, the top shows how these bytes are parsed as an MSDU, and the bottom shows how these bytes are parsed as an A-MSDU:



This shows that when an MSDU is turned into an A-MSDU by a malicious outsider, the destination address of the first subframe in the A-MSDU will be AA:AA:03:00:00:00 which equals the first 6 bytes of the RFC1042 header (specifically an LLC header followed by a SNAP header). Based on this, the submission “On A-MSDU addressing” added a defense to drop the A-MSDU if the destination address of the first A-MSDU subframe equals AA:AA:03:00:00:00: <https://mentor.ieee.org/802.11/dcn/21/11-21-0816-03-000mon-a-msdu-addressing.docx>

However, in a mesh BSS (MBSS), all MSDU frames start with a 6-byte Mesh Control field, followed by the RFC1042 header. The following figure shows example bytes of an MSDU that is sent by a mesh STA in a mesh BSS (MBSS), parsed as an MSDU (top), and when parsed as an A-MSDU (bottom):

Afbeelding met tekst, schermopname, Lettertype, nummer

Door AI gegenereerde inhoud is mogelijk onjuist.

If parsed as an MSDU (top), the frame starts with a 6-byte Mesh Control field, where the two least significant bits of the Flags subfield indicate the length of the optional Mesh Address Extension field, shown in yellow and bold, which is either 0, 6, or 12 bytes long.

When the same bytes are parsed as an A-MSDU in a MBSS (bottom), the destination field of the first A-MSDU does not equal AA-AA-03-00-00-00-00, meaning the previously-introduced defense does not work in an MBSS. Instead, to detect the attack, when a mesh STA receives an A-MSDU, the first byte should be parsed as the Flags field, and the 6-bytes at the offset where the RFC1042 header would start has to be compared against the value AA-AA-03-00-00-00-00 to detect the attack.

The above issue in an MBSS was confirmed with wpa\_supplicant 2.11 on Linux 6.12 using the mac80211\_hwsim driver: under the right conditions, a malicious outsider could abuse this to inject arbitrary frames into the MBSS. A proof-of-concept of the defense proposed in this submission was implemented on Linux kernel 6.1.110 and successfully detected when a malicious outsider changed the A-MSDU Present subfield to 1.

**Discussion of the defense**

Our proposed defense performs checks when M differs from 3, where M equals the two least significant bits of the first octet of the first A-MSDU subframe header. In case of an attack, these two bits equal the Address Extension Mode subfield in the Mesh Flags subfield of the 6-byte Mesh Control Field. Valid values for the Address Extension Mode are 0, 1, and 2, with the value 3 being a reserved value, see Table 9-35 in REVme/D7.0, and see the figure above for example frames. A legitimate STA is therefore not expected to use the value 3 since doing so would not be valid behavior. An adversary also cannot force this value to be 3, since that would cause integrity checks to fail during frame decryption. An adversary also cannot construct an encrypted frame themselves where M would equal 3, since we assume that the adversary is an outsider. We therefore only need to consider the values 0, 1, and 2 in the defense.

The proposed defense below has been adopted by Linux since kernel version 6.16, released on 27 July 2025. The commit adding the defense is 737bb912ebbe (“wifi: prevent A-MSDU attacks in mesh networks”) and can be viewed online at: <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=737bb912ebbe4571195c56eba557c4d7315b26fb>

**Proposed changes**

**10.11 A-MSDU operation**

***Editor: modify 10.11 (REVme/D7.0 P1983 L40) as indicated:***

* Each DA shall be the RA or, except if the A-MSDU is transmitted on a TDLS direct link, a group address, if none of the following conditions is met:
  + …
  + the frame is from a mesh STA

Otherwise, there is no restriction on each DA, except that if the frame is from a nonmesh STA, the first six octets of the first A-MSDU subframe header (the DA in a Basic A-MSDU) shall not be AA-AA-03-00-00-00.

NOTE 3—The address AA-AA-03-00-00-00 is that which results from an attack in which an encrypted QoS Data frame not containing an A-MSDU (whose unencrypted frame body therefore starts with an LLC header followed by a SNAP header constructed per IETF RFC 1042) has the A-MSDU Present subfield changed to 1 by an attacker to cause it to appear to be a payload protected A-MSDU (PP A-MSDU) with multiple MSDUs (each preceded by an A-MSDU subframe header, which starts with the DA).

***Editor: modify 10.11 (REVme/D7.0 P1984 L17), including the footnote, as indicated:***

* In an MBSS,
  + …
  + in a group addressed frame, each mesh DA, and the DA in the Address 1 field (see Table 9-77 (Address field contents for mesh Data and Multihop Action frames(#462)(#419))), shall be the same group address.
  + let *M* be the value of the two least significant bits of the first octet of the first A-MSDU subframe header, then if *M* is not equal to 3, the six octets at offset (6 + *M* × 6) measured from the first octet of the first A-MSDU subframe header shall not be AA-AA-03-00-00-00.

(#462)A STA that receives a frame containing an A-MSDU that violates these rules should discard it[[1]](#footnote-1).

**References:**

* 802.11me/D7.0
* [Fragment and Forge: Breaking Wi-Fi Through Frame Aggregation and Fragmentation](https://papers.mathyvanhoef.com/usenix2021.pdf). In USENIX Security Symposium, 2021. Permanently stored in ISBN 978-1-939133-24-3 at pages 161 to 178.

1. The check on the DA in the first A-MSDU subframe header for a nonmesh STA, and the offset-dependent check for a mesh STA, is important to defend against attacks from malicious outsiders when signaling and payload protected A-MSDUs (SPP A-MSDUs) are not being used. The other checks on the DAs and the checks on the SAs are important to defend against impersonation attacks from malicious insiders (which means there is no protection against malicious APs or PCPs, DMG or S1G relays, GLK STAs, and mesh STAs). The lack of checks on the SA and DA in the case of mesh and relay operation presents an attack surface that is outside the scope of remediation in this standard. In the case of GLK, use of a SYNRA for an individually addressed MSDU with multiple links indicated in the station vector parameter also presents such an attack surface, which can be avoided by transmitting multiple individually addressed MPDUs instead (see 10.64 (Addressing of GLK Data frame transmission)). [↑](#footnote-ref-1)