[<prev] [next>] [day] [month] [year] [list]

Date: Tue, 15 Jun 2021 22:33:18 +0200
From: Norbert Slusarek <nslusarek@....net>
To: oss-security@...ts.openwall.com
Cc: socketcan@...tkopp.net, mkl@...gutronix.de, menschel.p@...teo.de
Subject: CVE-2021-34693: Infoleak in CAN BCM protocol in Linux kernel

Hello,

this is an announcement for recently reported infoleaks in the CAN BCM networking protocol in the Linux kernel.

The vulnerability has been assigned CVE-2021-34693 and was found in kernels ranging from 2.6.25-rc1 to 5.12.10.

The infoleak can be found in struct bcm_msg_head, which is a structure used to describe CAN BCM messages. Due to an automatically introduced padding, the structure contains a 4-byte hole which is never initialized. The 4-byte hole will contain data from the kernel stack as the structure is allocated on the stack. Depending on the architecture, the leak happens at different places within the structure.

On 64-bit systems,
the 4-byte hole can be found between struct members count and ival1.
In this case, kernel addresses can be partially revealed.

On 32-bit systems,
the 4-byte hole can be found between struct members nframes and frames[0].
In this case, kernel addresses can be fully revealed, resulting in a feasible KASLR bypass.

The leak can be reached by an unprivileged user by reproducing the following steps:

- open and connect a CAN BCM socket
- sendmsg() with RX_SETUP on socket to setup CAN BCM message receiver
- message will be received by the message receiver, packed with struct bcm_msg_head and queued for reception
- recvmsg() to receive the message, finally leaking the uninitialized bytes to userspace

The patch can be found in the link below or in the attachments.
https://lore.kernel.org/netdev/trinity-87eaea25-2a7d-4aa9-92a5-269b822e5d95-1623609211076@3c-app-gmx-bs04/T/#me01c68ad3b6784f533f1b1509c95943bb5911457

A short PoC can be found in the link below or in the attachments.
https://github.com/nrb547/kernel-exploitation/tree/main/cve-2021-34693

Credits go out to Norbert Slusarek and Patrick Menschel.

View attachment "0001-fix-infoleak.patch**" of type "**text/x-patch**" (1851 bytes)**

View attachment "poc.c**" of type "**text/plain**" (1543 bytes)**