Products Services Publications Resources
What's new

Follow @Openwall on Twitter for new release announcements and other news

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

Message-ID: <818cb36f5cb471ae@sudo.ws>
Date: Mon, 14 Oct 2019 09:00:31 -0600
From: "Todd C. Miller" <Todd.Miller@...o.ws>
To: oss-security@...ts.openwall.com
Subject: Sudo: CVE-2019-14287

Sudo 1.8.28 has been today, October 14th, 2019 which includes a fix for the following security-related issue which has been assigned CVE-2019-14287. The information below is also available at https://www.sudo.ws/alerts/minus_1_uid.html

Potential bypass of Runas user restrictions

Summary:
When sudo is configured to allow a user to run commands as an arbitrary user via the ALL keyword in a Runas specification, it is possible to run commands as root by specifying the user ID -1 or 4294967295.

This can be used by a user with sufficient sudo privileges to run commands as root even if the Runas specification explicitly disallows root access as long as the ALL keyword is listed first in the Runas specification.

Log entries for commands run this way will list the target user as 4294967295 instead of root. In addition, PAM session modules will not be run for the command.

Sudo versions affected:
Sudo versions prior to 1.8.28 are affected.

CVE ID:
This vulnerability has been assigned CVE-2019-14287 in the Common Vulnerabilities and Exposures database.

Details:
Exploiting the bug requires that the user have sudo privileges that allow them to run commands with an arbitrary user ID. Typically, this means that the user's sudoers entry has the special value ALL in the Runas specifier.

Sudo supports running a command with a user-specified user name or user ID, if permitted by the sudoers policy. For example, the following sudoers entry allow the id command to be run as any user because it includes the ALL keyword in the Runas specifier.

```
    myhost alice = (ALL) /usr/bin/id
```

Not only is user "alice" is able to run the id command as any valid user, she is also able to run it as an arbitrary user ID by using the "#uid" syntax, for example:

```
    sudo -u#1234 id -u
```

would return 1234.

However, the setresuid(2) and setreuid(2) system calls, which sudo uses to change the user ID before running the command, treat user ID -1 (or its unsigned equivalent 4294967295), specially and do not change the user ID for this value. As a result,

```
    sudo -u#-1 id -u
```

or

```
    sudo -u#4294967295 id -u
```

will actually return 0. This is because the sudo command itself is already running as user ID 0 so when sudo tries to change to user ID -1, no change occurs.

This results in sudo log entries that report the command as being run by user ID 4294967295 and not root (or user ID 0). Additionally, because the user ID specified via the -u option does not exist in the password database, no PAM session modules will be run.

If a sudoers entry is written to allow the user to run a command as any user except root, the bug can be used to avoid this restriction. For example, given the following sudoers entry:

```
    myhost bob = (ALL, !root) /usr/bin/vi
```

User bob is allowed to run vi as any user but root. However, due to the bug, bob is actually able to run vi as root by running "sudo -u#-1 vi", violating the security policy.

Only sudoers entries where the ALL keyword is present in the Runas specifier are affected. For example, the following sudoers entry is unaffected:

```
    myhost alice = /usr/bin/id
```

In this example, alice is only allowed to run the id command as root. Any attempt to run the command as a different user will be denied.

Fix:
The bug is fixed in sudo 1.8.28.

Credit:
Joe Vennix from Apple Information Security found and analyzed the bug.

Patches:
See attached patch for sudo 1.8.27.