Dirty Pipe Cola

Agenda

- What the project is about
- What we did (technically speaking)
- How we did it
- Challenges and problems
- Things we couldn't do
- Future work
- Demo

What the project is about

PepsiCo and Coca-Cola Company have been competitors for ages. Pepsi wants to step on Coca-Cola once and for all, and hired some expert hackers, like us, to create a malware in order to spy on the other, by injecting it on their systems.

We used "social engineering" via e-mail to a group of Production managers that were invited for an urgent Zoom meeting because a critical production issue in their biggest manufacturing plant in US.

The main goal is to gain constant monitoring over the other company, in order to know their moves in advance and be able to overpass them businesswise.

The aims of this project:

- Phishing.
- Privilege escalation in the system.
- Open a backdoor for attacker.
- Automatic exfiltration of some data.
- Do not be detected by antiviruses on the workstation and IDS in the network of the company.
- Persistency.

Vulnerability - Dirty Pipe CVE-2022-0847

This vulnerability initially affects the **Linux kernel** from **version 5.8** onwards and **allows privilege escalation by writing to read-only locked files**. Many systems, including the latest versions of Android and some distributions such as Ubuntu, Debian or Fedora are affected.

Background terminology:

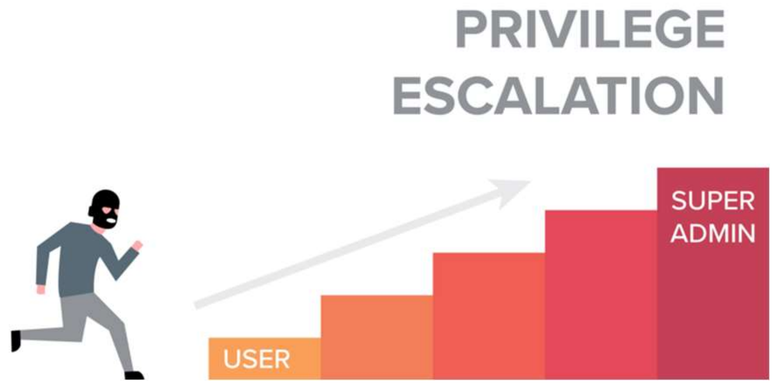
- **Pipe.** Is a data channel that can be used to communicate between two processes (inter-process communication).
- **Splice**. System call that allows to point a pipe at a page which is already loaded into memory.
- **Pipe flags.** Pipe flags specify characteristics such as status and permissions. It is in one of these flags, PIPE_BUF_FLAG_CAN_MERGE, where the problem arises

Origin of the vulnerability

- **Linux Kernel v4.9 (2016).** Allowing pipes to be created with arbitrary flags.
- **Linux Kernel v5.8 (2020).** Adding a new flag **PIPE_BUF_CAN_MERGE** that tells to the kernel that the page which is pointed by the pipe can be updated.

But since this **new flag** was defined, the lack of initialization did result in a problem. Since the kernel always has the cached pages under its control, **it does not check its permissions when using a page**.

Exploit: Local Privilege Escalation in Linux kernel

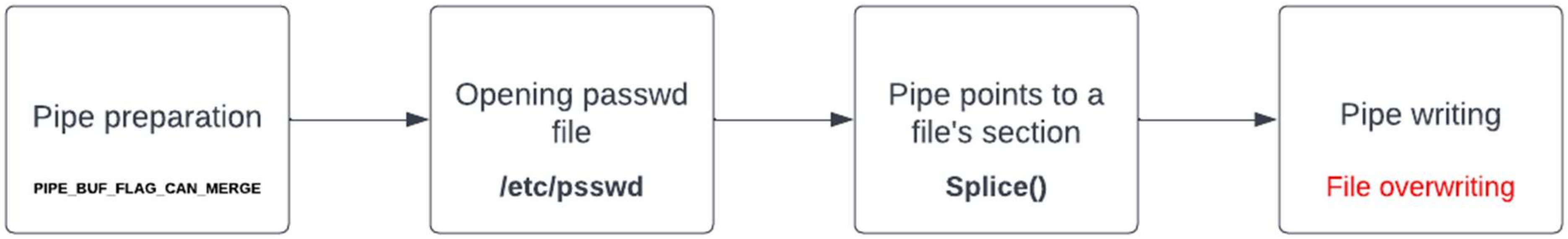


Date: 07/03/2022
Exploit Author: Max Kellermann <max.kellermann@ionos.com>
Tested on: ubuntu 20.04.1 LTS
Affect product:Linux kernel 5.8 or later
Fixed Product:Linux kernel 5.16.11, 5.15.25, 5.10.102
CVE ID: CVE-2022-0847

Timeline

- 2021-04-29: first support ticket about file corruption
- 2022-02-19: file corruption problem identified as Linux kernel bug, which turned out to be an exploitable vulnerability
- 2022-02-20: bug report, exploit and patch sent to the Linux kernel security team
- 2022-02-21: bug reproduced on Google Pixel 6; bug report sent to the Android Security Team
- 2022-02-21: patch sent to LKML (without vulnerability details) as suggested by Linus Torvalds, Willy Tarreau and Al Viro
- 2022-02-23: Linux stable releases with my bug fix (5.16.11, 5.15.25, 5.10.102)
- 2022-02-24: Google merges my bug fix into the Android kernel
- 2022-02-28: notified the linux-distros mailing list
- 2022-03-07: public disclosure

Exploit: Local Privilege Escalation in Linux kernel



What we did

1. Phishing attack by email, redirecting to a fake *zoom* website and asking to download an update.

1. When the victim launches the update, there's a local privilege escalation (CVE-2022-0847), using Dirty Pipe Exploit.

1. Background exfiltration of all files in the computer (from /home).

1. Backdoor service connected to attacker server, providing remote root access.

1. Anti-forensics techniques and persistency.

How we did it

- **Phishing attack:**

  1. Clone zoom website.
  2. Edit .html.
  3. Publish fake zoom website with a tricky Domain name “*zooom.us*”.
  4. Send an urgent meeting mail to the victim (with hidden link).

     (Perform attack)

  1. Send meeting cancelation mail to the victim.

How we did it

- **Malware:**

  1. Appears to be a zoom upgrade (**Trojan**).

     Background processes:

  1. Local escalation of privileges (**Dirty Pipe exploit**).

  1. Backdoor as systemd service, persistent and anti-failure. Allows remote root access from attacker server. Via bash's builtin **/dev/tcp** and **netcat**.

  1. Exfiltration of files. Using **curl**, everything in the machine is sent to attacker server.

How we did it

- **Obfuscation**

    1. Masking activity: **Changing MACE times** of files and services.
    2. Hide files: Service is placed on /etc/systemd/system, named as ***snapd.loading.service*** .
    3. Hide processes: /proc is remounted so **root processes are hidden** for unprivileged users.

Challenges and problems

- **First Vulnerability was patched [CVE-2021-3609].**
    - Solved [**CVE-2022-0847**]: More recent and easier to exploit vulnerability.
- **USB injection and auto-run.**
    - Solved: Changing scope performing a phishing attack.
- **Zoom update using a .deb file.**
    - Requires sudo privileges to install (which victims shouldn't have).
    - Solved: Creating a .tgz and script with friendly-user guide on README.

Things we couldn't do

- **Encrypting the payload.**
  - Encrypt a reverse shell and inject the payload afterwards.

- **Fooling debuggers.**
  - Ptrace wouldn’t work in a multi-thread program.
  - Execute before main is compiler dependant.

Future work

- Remove possible traces (Post exploitation).
- Anti-reverse and debugging techniques.
- Spread the infection through the internal network.
- Secure the phishing server (HTTPS).

Demo

Demo - Initial Setup

- Victim:
  - Ubuntu 20.04.4 (linux kernel 5.13.0).
  - Zoom client is installed.
  - DNS Mask on attacker's IP to zooom.us (Phishing simulation).

- Attacker:
  - Ready SimpleHTTPServerWithUpload on port 8080.
  - Ready Zoom fake server on port 80.
  - Listening on 9001 for remote shell.
  - Python 3.

Questions

BACKUP SLIDES

Backup Slide

Exploit: Local Privilege Escalation in Linux kernel

How It Works

1. It starts by opening a file in read mode, which can later be written to, even if the program does not have write permissions.
2. It creates a pipe with the pipe() system call. This function gives the same process access to the descriptors that allow writing and reading.
3. Writes any type of information into the pipe to fill it completely and have the memory pages flagged with the PIPE_BUF_FLAG_CAN_MERGE flag.
4. Once all the pages have been marked, it allows the kernel to free them by reading all the data from the pipe that it had written.
5. From this point on, when the kernel reserves memory pages making use of the functions introduced in 2016, it will not initialize their flags and they will be marked with the PIPE_BUF_FLAG_CAN_MERGE flag.
6. It uses the splice() function to load the file that had been opened at the beginning. The memory page allocated to this file will be the same as that of our empty pipe, because it was marked with the flag.
7. It directly overwrites the data in the pipe.

Our exploit can be used to elevate privileges on a system by overwriting the contents of /etc/passwd and change the root password to “piped”

Backup Slide

Vulnerability - Dirty Pipe CVE-2022-0847

This vulnerability initially affects the Linux kernel from version 5.8 onwards and allows privilege escalation by writing to read-only locked files. Many systems, including the latest versions of Android and some distributions such as Ubuntu, Debian or Fedora are affected.

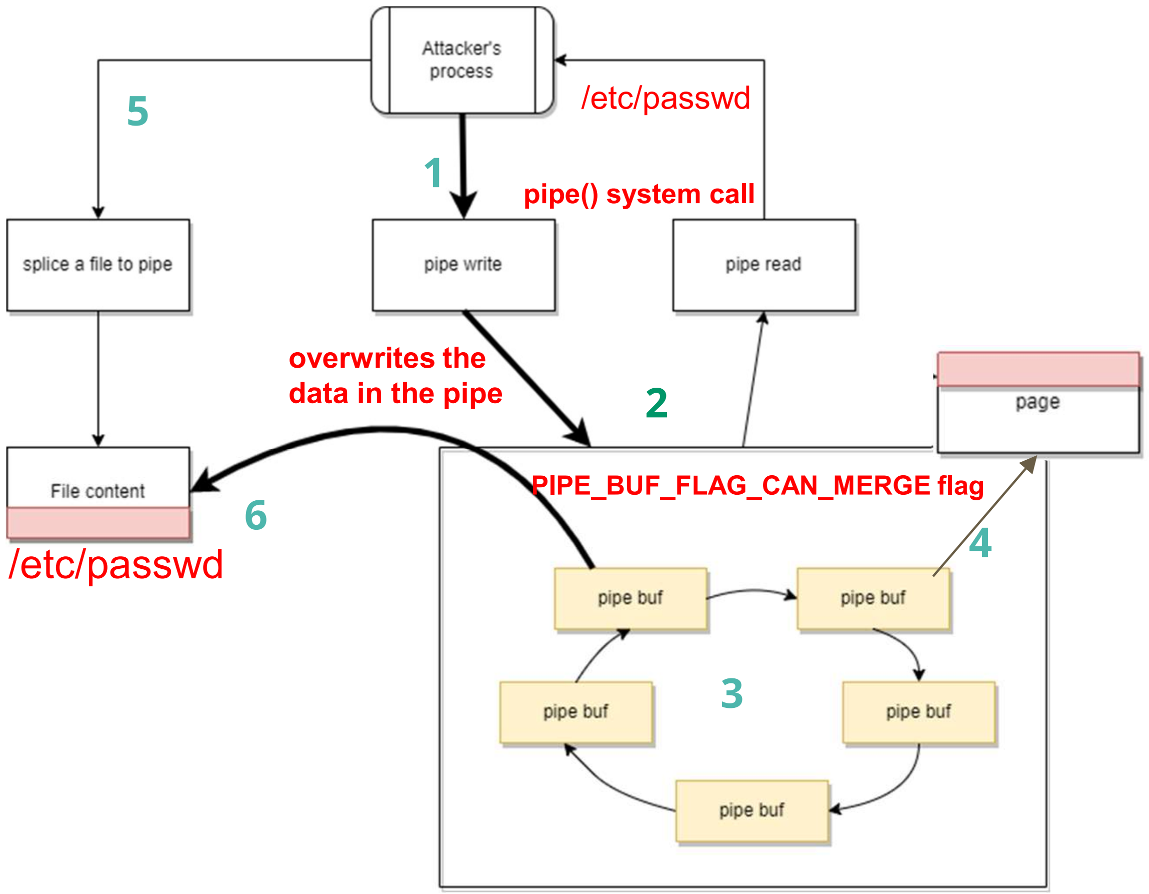
- **Pipes.** For inter-process communications, shared memory pages are often used, where one process reads and another writes. Typically, a pipe spans multiple memory pages. They are the same as those used for concatenating commands in the terminal, using the “|” character.
- **Anonymous pipes.** When the information shared between processes does not occupy an entire memory page, it can be reused for another pipe, resulting in data from different pipes coexisting in the same memory page.
- **Pipe flags.** Pipe flags specify characteristics such as status and permissions. It is in one of these flags, PIPE_BUF_FLAG_CAN_MERGE, where the problem arises

The **PIPE_BUF_FLAG_CAN_MERGE** flag was introduced in kernel 5.8, in a commit (f6dd975583bd8ce088400648fd9819e4691c8958) released in May 2020. Its function is to indicate that the data in a pipe of a page can be merged without the need to rewrite the data in memory.

But since this new flag was defined, the lack of initialization did result in a problem. Since the kernel always has the cached pages under its control, it does not check its permissions when using a page.

Backup Slide

Exploit: Local Privilege Escalation in Linux kernel



This exploit can be used to elevate privileges on a system by overwriting the contents of /etc/passwd and change the root password to “piped”.

To make this vulnerability more interesting, it not only works without write permissions, it also works with immutable files, on read-only btrfs snapshots and on read-only mounts (including CD-ROM mounts).

That is because the page cache is always writable (by the kernel), and writing to a pipe never checks any permissions.