Real Time Detection of Dirty Pipe (CVE-022-0847) Exploit

Project Proposal - Team B (Aaliyah, Katrina, Matthew & Xen)

Summary

In February 2022, a Linux zero day exploit was identified by security researcher Max Kellermann, dubbed Dirty Pipe (CVE-022-0847), which exploits an error in the pipe buffer flag handling, allowing a user with read-only access the ability to overwrite cached file pages including immutable files and read-only btrfs snapshots and mounts. The vulnerability “allows attackers to overwrite data in read-only files and SUID binaries to achieve root access”[1]. This exploitation can then be used for privilege escalation via code injection into root processes. Max Kellermann has since submitted a Linux kernel patch, and Android and other Linux and Linux-based distributions are in the process of pushing out patches[2].

The vulnerability has existed in the Linux kernel since 5.8, and official patches have been integrated into the Linux kernel versions 5.16.11, 5.15.25, and 5.10.102. Other versions of the kernel are not on current or long term support and have not received the patch. Google has also merged the fix into the Android operating system as of version 12-5.10.

Statement of Need

Given the complex threat landscape in today’s computing world, proactive cybersecurity evaluation of computer systems should be standard practice.

This exploit was discovered by a white hat hacker by log analysis and code review, but almost a year after the first customer reports of corrupted logs were reported. This exploitable vulnerability is now in the field and has since been publicly disclosed. Detection of this vulnerability is especially tricky due to the exploitation affecting the cached copy of files. When the device is power cycled or the kernel reclaims the memory of the page in the cache, the change is reverted, leaving no trace on the hard disk.

With consumers vulnerable, can defensive threat software detect the patterns of Dirty Pipe exploitation on vulnerable distributions in real time?

Project Activity, Methodology and Outcomes

This project is composed of three primary stages:

1. Setting up a testing lab with vulnerable target distributions for Dirty Pipe.

2. Executing the Dirty Pipe exploit in a consistently and easily reproducible way.

3. Determine if the dirty pipe exploit can be detected by defensive threat software in real time.

The penetration lab consists of three basic virtual machine types – two Kali Linux Oracle Virtual Appliances (OVA), and Ubuntu 20.04 LTS with Linux 5.8.0 kernel which is a vulnerable target Linux distribution; all these virtual machines will be connected on a single isolated network.

Network mapping and network scanning will be performed using one of the Kali VM's throughout the exploitation. Proof-of-concept testing will be performed from the other Kali VM on the Ubuntu VM to establish a protocol for penetration and exploitation. The exploitation demo will be written up as a small executable such that the exploit is easily and consistently reproduced.

Finally, various scanning and investigation techniques will be employed to determine if Dirty Pipe exploitation can be detected in real time by either the detection or the target machine.

Evaluation

Successful exploitation in the penetration testing and vulnerability scanning step will be benchmarked against the reports filed as part of the report for the CVE database.

A successful real-time detection will be considered successful if either the vulnerable target machine or the detection machine are able to accurately detect the Dirty Pipe exploitation before overwritten cached files are accessed by the kernel. If time permits, potential interventions and mitigations to defend against Dirty Pipe exploitations will be identified that owners of vulnerable kernels can implement in the interim before updating their distributions.

Dissemination

The results of the paper will be shared to the class (CU Boulder CSCI5403 Cybersecurity Fall 2022) and with our Professor, Piotr Windya.

References

1. Kellermann, Max. “The Dirty Pipe Vulnerability¶.” The Dirty Pipe Vulnerability - The Dirty Pipe Vulnerability Documentation, CM4All, https://dirtypipe.cm4all.com/.

2. Simpson, Michael T. Hands-on Ethical Hacking and Network Defense. Wadsworth Publishing Company, 2012.

3. Zorz, Zeljka. “Easily Exploitable Linux Bug Gives Root Access to Attackers (CVE-2022-0847).” Help Net Security, HelpNetSecurity, 8 Mar. 2022, https://www.helpnetsecurity.com/2022/03/08/cve-2022-0847/.

4. “CVE-2022-0847.” CVE, 3 Mar. 2022, https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-0847.