New York Tech - Vancouver
06-09-2025

Project Proposal: CVE-2022-0847 — Simulating the Dirty Pipe Linux Kernel Privilege Escalation Exploit in a Controlled Lab Environment

Team: Hackstreet Boys

Members: Humberto Da Silva Goncalves
John Jiang
Folorunso Kolapo

BACKGROUND

Privilege escalation remains a critical threat vector in modern Linux environments, especially in post-compromise scenarios. CVE-2022-0847, dubbed "Dirty Pipe," is a high-impact local privilege escalation vulnerability affecting Linux Kernel versions 5.8 and later. The flaw resides in the way the pipe buffer structure handles flags, allowing an unprivileged local user to overwrite arbitrary data in read-only files, including /etc/passwd or binary executables.

This project will simulate the Dirty Pipe exploit within a controlled Kali Linux or Debian-based sandbox to replicate a real-world post-exploitation attack chain. The operation will include forensic artifact analysis, privilege escalation verification, and evaluation of defensive mechanisms such as Kernel lockdown mode, AppArmor, and read-only filesystem protections. This demonstration provides critical insights into kernel-level vulnerabilities and their exploitation in modern Linux systems.

OBJECTIVES

- Reproduce CVE-2021-0847 in an isolated lab to understand exploit mechanics.
- Map the kill chain using the MITRE ATT&CK framework, aligning with T1068 (Exploitation for Privilege Escalation).
- Document forensic traces before, during, and after escalation for detection signature creation.
- Demonstrate mitigations including OS patching, AppArmor policy reinforcement, and EDR detection mapping.

SCOPE

- Simulate the exploit on vulnerable Linux distributions (Ubuntu 20.04, Debian).
- Monitor all I/O, process behaviour, and system logs using auditd, Sysmon for Linux, and custom bash wrappers.
- Visualize process tree and payload execution using tools like psrecord, gdb, and strace.
- Propose hardened configurations and showcase failed exploit attempts under patched systems.

METHODOLOGY

1. Lab Environment Setup

- Set up an isolated virtual environment using Kali Linux, Ubuntu, or Debian as the target system.
- Ensure the Linux kernel version is vulnerable (between 5.8 and 5.16.11).
- Disable automatic updates to prevent vulnerability patching.
- Configure the network to host-only or NAT to ensure full containment of the test environment.

2. Acquire and Prepare the Exploit

- Download a trusted public proof-of-concept (PoC) for Dirty Pipe from a security researcher or GitHub repository.
- Compile the exploit using a standard C compiler available on the Linux system.
- Verify that the target file for exploitation is writable under the vulnerability conditions.

3. Identify Exploitation Target

- Select a high-value target file, such as /etc/passwd, which allows privilege escalation when modified.
- Make a backup copy of the target file for restoration after the test.
- Determine the payload or user entry to be injected (e.g., a new root-equivalent user).

4. Exploit Execution

- Execute the compiled Dirty Pipe exploit against the chosen file.
- Inject the malicious modification into the file to create unauthorized root access.
- Attempt to switch to the elevated user account to confirm the exploit was successful.

5. Post-Exploitation Verification

- Validate that root-level access has been achieved.
- Review system behaviour and stability post-exploit.
- Analyze changes made to the exploited file and compare with the original backup.

6. MITRE ATT&CK Mapping

- Map the activity to relevant MITRE ATT&CK tactics and techniques:
- Privilege Escalation (T1068)
- Defense Evasion (T1222: File Permission Modification)

7. Detection and Mitigation Analysis

- Evaluate if the exploit generated system logs or alerts using tools like journalctl, auditd, or Sysmon for Linux.
- Demonstrate how applying vendor patches or kernel updates mitigates the vulnerability.
- Test additional hardening techniques such as AppArmor, SELinux, or kernel lockdown mode to restrict exploitability.

8. Environment Restoration

- Restore the compromised file from its backup to revert the system to its pre-exploit state.

TIMELINE

Phase	Description of Work	Start and End Dates
One	Research CVE, set up testbed (Kali, Ubuntu)	June 12–14, 2025
Two	Exploitation, logging, and forensic data collection	June 15–18, 2025
Three	Documentation, mitigation testing, and report generation	June 19–22, 2025

New York Tech - Vancouver
06-09-2025

Project Proposal: CVE-2022-0847 — Simulating the Dirty Pipe Linux Kernel Privilege Escalation Exploit in a Controlled Lab Environment

Team: Hackstreet Boys

Members: Humberto Da Silva Goncalves
John Jiang
Folorunso Kolapo

RESPONSIBILITIES

Tasks	Humberto Goncalves	John Jian	Folorunso Kolapo
Task 1	Execute the exploit	Set up Lab Environment	Execute the exploit
Task 2	Record Exploit Demo	Acquire and Build Exploit	Configure Monitoring Tools
Task 3	Map to MITRE ATT&CK	Prepare Exploitation Target	Analyze forensic artifacts
Task 4	Write Final Report	Execute the Exploit	Apply and test mitigations

MONITORING AND EVALUATION

Progress will be tracked against the following key indicators:

- Exploit reproducibility: Proof of successful privilege escalation.
- Log integrity: Comprehensive logs showing attack vectors and system reaction.
- Mitigation effectiveness: Demonstration that the patched system resists exploit reliably.
- Reporting quality: Technical documentation and visuals' clarity, depth, and reproducibility.

Peer review will be sought from classmates; checkpoints will be validated via instructor feedback and lab replays.

Approval Signature

____
Sara Khanchi, Professor