****

**Host and Network Security**

**Group Project**

Due Date: Friday the 21st of April, 2017

Student: Hugh Mullins, Student ID: 16106989

Student: Sunday Jimoh, Student ID: 16090462

Student: Conor McDonnel, Student ID: 16120604

Student: Raksha Ramdas, Student ID: 16052439

Student: Samaa Alsafwani, Student ID: 16087941

Course: Information and Network Security MENG

Department: Electronic & Computer Engineering

**Table of Contents**

[1 Dirty Copy-on-Write (CVE-2016-5195) 3](#_Toc480038885)

[1.1 Introduction 3](#_Toc480038886)

[1.2 Impacts of Exploiting the Dirty COW Vulnerability 3](#_Toc480038887)

[1.3 How to execute a Dirty COW Exploit 4](#_Toc480038888)

# Dirty Copy-on-Write (CVE-2016-5195)

## Introduction

Dirty copy-on-write (Dirty COW) is a security vulnerability present in the Linux kernel since version 2.6.22 which was released in July, 2007, Its CVE designation is CVE-2016-5195. It affects all major vendors of Linux including Ubuntu, Debian, Red Hat and even Android, which is also powered by the Linux kernel. It was discovered by a Linux security researcher, Phil Oester.

Copy-on-write (COW), sometimes referred to as implicit sharing or shadowing, manages memory resources and allows for more than one process to share a page until a user writes to the page, this is known in programming as marking a page dirty.

Dirty COW is a local privilege escalation bug that exploits a race condition in mm/gup.c in the Linux kernel version 2.x through 4.x, considering the way the Linux kernel’s memory subsystem handles the copy-on-write (COW) breakage of private read-only memory mappings. An unprivileged user or a local attacker could exploit this flaw to gain write access to otherwise read-only memory mappings and thus increase their privileges on the system. This vulnerability could be used by an attacker to modify existing setuid files with instructions to gain administrative privileges.

## Impacts of Exploiting the Dirty COW Vulnerability

* It allows attackers with local system accounts to modify on-disk binaries, bypassing the standard permission mechanisms that would prevent modification without an appropriate permission set.
* Because of the race condition, with the right timing, an unprivileged or a local attacker could use this flaw to gain write access to otherwise read-only memory mappings and thus increase their privileges on the system.
* Although it is a local privilege escalation bug, it can be used by remote attackers in conjunction with other known exploits that allow remote execution of non-privileged code to achieve remote root access on an affected computer. The attack itself does not leave any traces in the system log.
* It can also be used to obtain root permissions in any Android device up to Android version 7.

## How to execute a Dirty COW Exploit