01.NX Bit(MS : DEP)

List

NX Bit(MS : DEP)

- **NX Bit(NX bit, Never eXecute bit, 실행 방지 비트)이란?**
  - 프로세스 명령어나 코드 또는 데이터 저장을 위한 메모리 영역을 따로 분리하는 CPU의 기술입니다.
  - NX 특성으로 지정된 모든 메모리 구역은 데이터 저장을 위해서만 사용되며, 프로세서 명령어가 그 곳에 상주하지 않음으로써 실행되지 않도록 만들어 준다.
- **DEP(Data Execution Prevention)이란?**
  - 마이크로소프트 윈도우 운영 체제에 포함된 보안 기능이며, 악의적인 코드가 실행되는 것을 방지하기 위해 메모리를 추가로 확인하는 하드웨어 및 소프트웨어 기술입니다.
  - DEP는 두 가지 모드로 실행된다.
    - 하드웨어 DEP: 메모리에 명시적으로 실행 코드가 포함되어 있는 경우를 제외하고 프로세스의 모든 메모리 위치에서 실행할 수 없도록 표시합니다.
      - 대부분의 최신 프로세서는 하드웨어 적용 DEP를 지원합니다.
    - 소프트웨어 DEP: CPU가 하드웨어 DEP를 지원하지 않을 경우 사용합니다.
- **예를 들어 공격자가 Heap, Stack 영역에 Shellcode를 저장해서 실행하기 위해서는 해당 영역에 실행권한이 있어야 합니다.**
  - DEP가 적용되지 않았을 경우에는 쉘코드가 실행이 됩니다.
  - DEP가 적용된 경우에는 실행권한이 없으므로 쉘코드가 실행되지 않습니다.
    - 프로그램에서 해당 동작에 대한 예외처리 후 프로세스가 종료가 됩니다.

Example program

- bof 취약점이 존재하는 프로그램에 빌드 할 때 스택에 실행권한을 설정하여 컴파일 합니다.

DEP

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    char str[256];
    char *chare = (char*)malloc(100);

    printf("Input: ");
    gets(str);
    printf("%p\n", str);
}
```

Build Command(DEP disable)

gcc -z execstack -o DEP-disabled DEP.c

Check the protection techniques of binary files.

checksec.sh

- **Checksec.sh에서 다음과 같은 결과를 출력합니다.**
    - DEP-disabled file: NX disabled
    - DEP-enabled file: NX enabled

DEP disabled	**gcc -z execstack -o DEP-disabled DEP.c**
	`lazenca0x0@ubuntu:~/Documents/Definition/protection$ checksec.sh --file DEP-disabled`
	`RELRO  STACK CANARY  NX  PIE  RPATH  RUNPATH  FILE`
	`Partial RELRO  Canary found  NX disabled  No PIE  No RPATH  No RUNPATH  DEP-disabled`

DEP enabled	**gcc -o DEP-enabled DEP.c**
	`lazenca0x0@ubuntu:~/Documents/Definition/protection$ checksec.sh --file DEP-enabled`
	`RELRO  STACK CANARY  NX  PIE  RPATH  RUNPATH  FILE`
	`Partial RELRO  Canary found  NX enabled  No PIE  No RPATH  No RUNPATH  DEP-enabled`

Checking Permissions in Memory

- **다음과 같이 메모리 맵에서 메모리 영역별 설정된 권한을 확인할 수 있습니다.**
    - DEP enabled의 경우 실행권한(--x-)을 가지고 있는 영역은 5곳입니다.
    - DEP disabled의 경우 실행권한(--x-)을 가지고 있는 영역은 17곳입니다.

DEP enabled

```
lazenca0x0@ubuntu:~$ cat /proc/6339/maps
00400000-00401000 r-xp 00000000 08:01 424692                             /home/lazenca0x0/Documents
/Definition/protection/DEP-enabled
00600000-00601000 r--p 00000000 08:01 424692                             /home/lazenca0x0/Documents
/Definition/protection/DEP-enabled
00601000-00602000 rw-p 00001000 08:01 424692                             /home/lazenca0x0/Documents
/Definition/protection/DEP-enabled
01e10000-01e31000 rw-p 00000000 00:00 0                                  [heap]
7fe1b704c000-7fe1b720c000 r-xp 00000000 08:01 655589                     /lib/x86_64-linux-gnu/libc-
2.23.so
7fe1b720c000-7fe1b740c000 ---p 001c0000 08:01 655589                     /lib/x86_64-linux-gnu/libc-
2.23.so
7fe1b740c000-7fe1b7410000 r--p 001c0000 08:01 655589                     /lib/x86_64-linux-gnu/libc-
2.23.so
7fe1b7410000-7fe1b7412000 rw-p 001c4000 08:01 655589                     /lib/x86_64-linux-gnu/libc-
2.23.so
7fe1b7412000-7fe1b7416000 rw-p 00000000 00:00 0
7fe1b7416000-7fe1b743c000 r-xp 00000000 08:01 655548                     /lib/x86_64-linux-gnu/ld-
2.23.so
7fe1b761c000-7fe1b761f000 rw-p 00000000 00:00 0
7fe1b7639000-7fe1b763b000 rw-p 00000000 00:00 0
7fe1b763b000-7fe1b763c000 r--p 00025000 08:01 655548                     /lib/x86_64-linux-gnu/ld-
2.23.so
7fe1b763c000-7fe1b763d000 rw-p 00026000 08:01 655548                     /lib/x86_64-linux-gnu/ld-
2.23.so
7fe1b763d000-7fe1b763e000 rw-p 00000000 00:00 0
7ffc8bf50000-7ffc8bf71000 rw-p 00000000 00:00 0                          [stack]
7ffc8bfc7000-7ffc8bfc9000 r--p 00000000 00:00 0                          [vvar]
7ffc8bfc9000-7ffc8bfcb000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
lazenca0x0@ubuntu:~$
```

DEP disabled

```
lazenca0x0@ubuntu:~$ cat /proc/6422/maps
00400000-00401000 r-xp 00000000 08:01 424690                             /home/lazenca0x0/Documents
/Definition/protection/DEP-disabled
00600000-00601000 r-xp 00000000 08:01 424690                             /home/lazenca0x0/Documents
/Definition/protection/DEP-disabled
00601000-00602000 rwxp 00001000 08:01 424690                             /home/lazenca0x0/Documents
/Definition/protection/DEP-disabled
023f8000-02419000 rwxp 00000000 00:00 0                                  [heap]
7f9c009e4000-7f9c00ba4000 r-xp 00000000 08:01 655589                     /lib/x86_64-linux-gnu/libc-
2.23.so
7f9c00ba4000-7f9c00da4000 ---p 001c0000 08:01 655589                     /lib/x86_64-linux-gnu/libc-
2.23.so
7f9c00da4000-7f9c00da8000 r-xp 001c0000 08:01 655589                     /lib/x86_64-linux-gnu/libc-
2.23.so
7f9c00da8000-7f9c00daa000 rwxp 001c4000 08:01 655589                     /lib/x86_64-linux-gnu/libc-
2.23.so
7f9c00daa000-7f9c00dae000 rwxp 00000000 00:00 0
7f9c00dae000-7f9c00dd4000 r-xp 00000000 08:01 655548                     /lib/x86_64-linux-gnu/ld-
2.23.so
7f9c00fb4000-7f9c00fb7000 rwxp 00000000 00:00 0
7f9c00fd1000-7f9c00fd3000 rwxp 00000000 00:00 0
7f9c00fd3000-7f9c00fd4000 r-xp 00025000 08:01 655548                     /lib/x86_64-linux-gnu/ld-
2.23.so
7f9c00fd4000-7f9c00fd5000 rwxp 00026000 08:01 655548                     /lib/x86_64-linux-gnu/ld-
2.23.so
7f9c00fd5000-7f9c00fd6000 rwxp 00000000 00:00 0
7ffed60cf000-7ffed60f0000 rwxp 00000000 00:00 0                          [stack]
7ffed61c5000-7ffed61c7000 r--p 00000000 00:00 0                          [vvar]
7ffed61c7000-7ffed61c9000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
lazenca0x0@ubuntu:~$
```

How to detect NX in the "Checksec.sh" file

Binary

- **다음과 같은 방법으로 바이너리의 NX 설정여부를 확인합니다.**
    - readelf 명령어를 이용해 파일의 세그먼트 헤더 정보에서 NX 여부를 확인합니다.
    - 파일의 세그먼트 헤더 정보에서 'GNU_STACK'의 Flg 값이 'RWE'이라면 NX가 활성화되었다고 판단합니다.

Checksec.sh - line 163

```
  # check for NX support
  if readelf -W -l $1 2>/dev/null | grep 'GNU_STACK' | grep -q 'RWE'; then
    echo -n -e '\033[31mNX disabled\033[m   '
  else
    echo -n -e '\033[32mNX enabled \033[m   '
  fi
```

- NX가 적용된 바이너리의 Flg 값이 'RW' 입니다.
- NX가 적용되지 않은 바이너리의 Flg 값이 'RWE' 입니다.

readelf -W -l ./DEP-disabled |grep 'GNU_STACK' | grep 'RWE'

```
lazenca0x0@ubuntu:~/Documents/Definition/protection$ readelf -W -l ./DEP-disabled |grep 'GNU_STACK'
  GNU_STACK      0x000000 0x0000000000000000 0x0000000000000000 0x000000 0x000000 RWE 0x10
lazenca0x0@ubuntu:~/Documents/Definition/protection$ readelf -W -l ./DEP-disabled |grep 'GNU_STACK' | grep 'RWE'
  GNU_STACK      0x000000 0x0000000000000000 0x0000000000000000 0x000000 0x000000 RWE 0x10
lazenca0x0@ubuntu:~/Documents/Definition/protection$
```

readelf -W -l ./DEP-enabled |grep 'GNU_STACK' | grep 'RWE'

```
lazenca0x0@ubuntu:~/Documents/Definition/protection$ readelf -W -l ./DEP-enabled |grep 'GNU_STACK'
  GNU_STACK      0x000000 0x0000000000000000 0x0000000000000000 0x000000 0x000000 RW  0x10
lazenca0x0@ubuntu:~/Documents/Definition/protection$ readelf -W -l ./DEP-enabled |grep 'GNU_STACK' | grep 'RWE'
lazenca0x0@ubuntu:~/Documents/Definition/protection$
```

Process

- **다음과 같은 방법으로 실행중인 프로세서의 NX 설정여부를 확인합니다.**
    - Binary의 확인 방식과 동일하며, 전달되는 파일의 경로가 다음과 같이 다릅니다.
        - Ex) /proc/<PID>/exe

Checksec.sh - line 249

```
  # fallback check for NX support
  elif readelf -W -l $1/exe 2>/dev/null | grep 'GNU_STACK' | grep -q 'RWE'; then
    echo -n -e '\033[31mNX disabled\033[m   '
  else
    echo -n -e '\033[32mNX enabled \033[m   '
  fi
```

readelf -W -l /proc/<PID>/exe |grep 'GNU_STACK'

```
lazenca0x0@ubuntu:~/Documents/Definition/protection$ ps -ef|grep DEP
lazenca+   6586   6369  0 20:22 pts/18   00:00:00 ./DEP-disabled
lazenca+   6607   6173  0 20:23 pts/4    00:00:00 grep --color=auto DEP
lazenca0x0@ubuntu:~/Documents/Definition/protection$ readelf -W -l /proc/6586/exe |grep 'GNU_STACK'
  GNU_STACK      0x000000 0x0000000000000000 0x0000000000000000 0x000000 0x000000 RWE 0x10
lazenca0x0@ubuntu:~/Documents/Definition/protection$ readelf -W -l /proc/6586/exe |grep 'GNU_STACK' | grep 'RWE'
  GNU_STACK      0x000000 0x0000000000000000 0x0000000000000000 0x000000 0x000000 RWE 0x10
lazenca0x0@ubuntu:~/Documents/Definition/protection$
```

CPU

- **다음과 같은 방법으로 CPU의 NX 설정여부를 확인합니다.**
  - "/proc/cpuinfo" 파일에서 'nx' 문자가 있는지 확인합니다.

Checksec.sh - line 324

```
# check cpu nx flag
nxcheck() {
  if grep -q nx /proc/cpuinfo; then
    echo -n -e '\033[32mYes\033[m\n\n'
  else
    echo -n -e '\033[31mNo\033[m\n\n'
  fi
}
```

grep nx /proc/cpuinfo

```
lazenca0x0@ubuntu:~/Documents/Definition/protection$ grep nx /proc/cpuinfo
flags                : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts mmx
fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts nopl xtopology tsc_reliable
nonstop_tsc aperfmperf eagerfpu pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt
tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm epb fsgsbase tsc_adjust bmi1 avx2 smep bmi2
invpcid xsaveopt dtherm ida arat pln pts
lazenca0x0@ubuntu:~/Documents/Definition/protection$
```

Related information

- https://support.microsoft.com/ko-kr/help/912923/how-to-determine-that-hardware-dep-is-available-and-configured-on-your