06.PIE

Unknown macro: 'html'

List

PIE(Position Independent Executable)

Description

- PIE(Position Independent Executable) .

Example

Source code

```
#include <stdio.h>

char *gBuf = "Lazenca.0x0";

void lazenca() {
        printf("Lazenca.0x1\n");
}

void main(){
        printf("[.data]    : %p\n",gBuf);
        printf("[Function] : %p\n",lazenca);
}
```

Build PIE file

```
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ gcc -o NoPIE PIE.c
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ gcc -fPIE -pie -o PIE PIE.c
```

Check the protection techniques of binary files.

checksec.sh

- **Checksec.sh** .

- NoPIE : "No PIE"
- PIE : "PIE enabled"

checksec.sh --file NoPIE

```
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ checksec.sh --file NoPIE
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      FILE
Partial RELRO   No canary found   NX enabled    No PIE          No RPATH   No RUNPATH   NoPIE
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$
```

checksec.sh --file PIE

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

Compare PIE and NonPIE.

Address

- **PIE .**
  - PIE .

NoPIE

```
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ ./NoPIE
[.data]    : 0x400634
[Function] : 0x400566
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ ./NoPIE
[.data]    : 0x400634
[Function] : 0x400566
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ ./NoPIE
[.data]    : 0x400634
[Function] : 0x400566
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$
```

- **PIE .**
  - PIE .

PIE

```
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ ./PIE
[.data]    : 0x563d12821884
[Function] : 0x563d128217b0
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ ./PIE
[.data]    : 0x55cbbaae3884
[Function] : 0x55cbbaae37b0
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ ./PIE
[.data]    : 0x55f7c9a1e884
[Function] : 0x55f7c9a1e7b0
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$
```

Code

- **PIE .**
  - PIE .
  - PIE offset .
    - offset .
  - .
    - (0x555555554000) + .text main offset (0x7c3) = main (0x00005555555547c3)

NoPIE

```
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ gdb -q ./NoPIE
Reading symbols from ./NoPIE...(no debugging symbols found)...done.
gdb-peda$ disassemble main
Dump of assembler code for function main:
   0x0000000000400577 <+0>:     push   rbp
   0x0000000000400578 <+1>:     mov    rbp,rsp
   0x000000000040057b <+4>:     mov    rax,QWORD PTR [rip+0x200abe]        # 0x601040 <gBuf>
   0x0000000000400582 <+11>:     mov    rsi,rax
   0x0000000000400585 <+14>:     mov    edi,0x40064c
   0x000000000040058a <+19>:     mov    eax,0x0
   0x000000000040058f <+24>:     call   0x400440 <printf@plt>
   0x0000000000400594 <+29>:     mov    esi,0x400566
   0x0000000000400599 <+34>:     mov    edi,0x40065d
   0x000000000040059e <+39>:     mov    eax,0x0
   0x00000000004005a3 <+44>:     call   0x400440 <printf@plt>
   0x00000000004005a8 <+49>:     nop
   0x00000000004005a9 <+50>:     pop    rbp
   0x00000000004005aa <+51>:     ret
End of assembler dump.
gdb-peda$
```

PIE

```
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ gdb -q ./PIE
Reading symbols from ./PIE...(no debugging symbols found)...done.
gdb-peda$ disassemble main
Dump of assembler code for function main:
   0x00000000000007c3 <+0>:     push   rbp
   0x00000000000007c4 <+1>:     mov    rbp,rsp
   0x00000000000007c7 <+4>:     mov    rax,QWORD PTR [rip+0x200872]        # 0x201040 <gBuf>
   0x00000000000007ce <+11>:     mov    rsi,rax
   0x00000000000007d1 <+14>:     lea    rdi,[rip+0xc4]        # 0x89c
   0x00000000000007d8 <+21>:     mov    eax,0x0
   0x00000000000007dd <+26>:     call   0x650 <printf@plt>
   0x00000000000007e2 <+31>:     lea    rsi,[rip+0xfffffffffffffc7]        # 0x7b0 <lazenca>
   0x00000000000007e9 <+38>:     lea    rdi,[rip+0xbd]        # 0x8ad
   0x00000000000007f0 <+45>:     mov    eax,0x0
   0x00000000000007f5 <+50>:     call   0x650 <printf@plt>
   0x00000000000007fa <+55>:     nop
   0x00000000000007fb <+56>:     pop    rbp
   0x00000000000007fc <+57>:     ret
End of assembler dump.
gdb-peda$
```

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

Binary

- **Canary .**
  - 'readelf' ELF Header PIE .
  - "Type:" "EXEC" PIE .
  - "Type:" "DYN" PIE .
    - 'readelf' "Dynamic section" "DEBUG" section PIE .

Checksec.sh - line 170

```
  # check for PIE support
  if readelf -h $1 2>/dev/null | grep -q 'Type:[[:space:]]*EXEC'; then
    echo -n -e '\033[31mNo PIE       \033[m   '
  elif readelf -h $1 2>/dev/null | grep -q 'Type:[[:space:]]*DYN'; then
    if readelf -d $1 2>/dev/null | grep -q '(DEBUG)'; then
      echo -n -e '\033[32mPIE enabled  \033[m   '
    else
      echo -n -e '\033[33mDSO          \033[m   '
    fi
  else
    echo -n -e '\033[33mNot an ELF file\033[m   '
  fi
```

NoPIE

```
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ readelf -h ./NoPIE |grep 'Type:[[:space:]]*EXEC'
  Type:                              EXEC (Executable file)
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$
```

PIE

```
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ readelf -h ./PIE |grep 'Type:[[:space:]]*DYN'
  Type:                              DYN (Shared object file)
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$ readelf -d ./PIE |grep '(DEBUG)'
 0x0000000000000015 (DEBUG)              0x0
lazenca0x0@ubuntu:~/Documents/Definition/protection/PIE$
```

Process

- **Canary .**
  - Binary , .
    - Ex) /proc/<PID>/exe

Checksec.sh - line 256

```
  # check for PIE support
  if readelf -h $1/exe 2>/dev/null | grep -q 'Type:[[:space:]]*EXEC'; then
    echo -n -e '\033[31mNo PIE               \033[m   '
  elif readelf -h $1/exe 2>/dev/null | grep -q 'Type:[[:space:]]*DYN'; then
    if readelf -d $1/exe 2>/dev/null | grep -q '(DEBUG)'; then
      echo -n -e '\033[32mPIE enabled          \033[m   '
    else
      echo -n -e '\033[33mDynamic Shared Object\033[m   '
    fi
  else
    echo -n -e '\033[33mNot an ELF file      \033[m   '
  fi
```

Related information

- N/a

Unknown macro: 'html'