God'Dragon
EverTokki@LeaveRet
evertokki.tistory.com

==========

Failure to Boot
Score: 20

> After opening the robot's front panel and looking inside, you discover a small red button behind a tangle of wires. Pressing the button lights up the robot's primary screen. It glows black and quickly flashes blue. A line of small text types out:
>
> > ERROR: 0x00000023
>
> The text refreshes and displays the prompt:
>
> > FILE SYSTEM RECOVERY INITIATED...
> > FILE SYSTEM COULD NOT BE IDENTIFIED...
> > PLEASE ENTER FILE SYSTEM FORMAT:

-에러메세지를 구글에 쳐봅니다. 첫 검색결과 내용을 보면:

> Stop error code 0x00000023 FAT_FILE_SYSTEM -
> Microsoft Support: support.microsoft.com/kb/900626

-FAT_ FILE_SYSTEM이란 것을 볼 수 있습니다.

Key: Fat

Grep is Your Friend
Score: 40

> After plugging the robot into the computer, the robot asks for the name of a file containing the string SECRET AUTH CODE. You can find it using the command-line interface in /problems/grep.tar or by downloading all of the files.

-Administratorui-MacBook-Pro:grep EverTokki$ grep SECRET *
fHYYpdrfeOCHyQicfe96xfw==:SECRET AUTH CODES
grep로 해당 문자열이 들어있는 파일을 찾습니다.

Key: fHYYpdrfeOCHyQicfe96xfw==

XMLOL
Score: 30

> The book has instructions on how to dump the corrupted configuration file from the robot's memory. You find a corrupted XML file and are looking for a configuration key.

https://picoctf.com/autoproblems/tmpQFNbqS.xml

-XML파일을 보면 아무것도 없어요. 그렇다면 소스보기를 합니다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<garbage
 <writing>
  <?xml verion="1.0" encoding"UTF-8"
     is really

    <super_secret_flag>284873429490034423589631455</super_secret_flag>

</ gar <bage>
```

-<super_secret_flag>284873429490034423589631455</super_secret_flag>가 눈에 띕니다

Key: 284873429490034423589631455

Read the Manual
Score: 30

> On the back of the broken panel you see a recovery manual. You need to find the emergency repair key in order to put the robot into autoboot mode, but it appears to be ciphered using a Caesar cipher.

https://picoctf.com/autoproblems/tmpA9_4Xm.txt

-RTFW!!!!!!!!!!!!RRRRTTTTFFFWWWWWWWW
…아닙니다. 암튼 열어보면 외계어가 써있습니다. 으음.. 그렇다면 해독해봐야겠죠? 구글에 auto caesar cipher decoder를 검색해봅니다. Caesar cipher이란 글전체에 있는 알파벳을 n자리만큼씩 옮겨 암호화한 것입니다. 고로 이건 다른 홈페이지 가서 수동으로 디코딩 해도 되는데 요즘 세상이 좋아서 그럴필요 없고 그냥 이 사이트에 들어가서 자동으로 컴퓨터가 가장 알맞는 디코딩을 해줍니다.

http://nayuki.eigenstate.org/page/automatic-caesar-cipher-breaker-javascript

-디코딩 하면 한 문자열이 눈에 띕니다.

> IMPORTANT: To enter automatic recovery mode, enter the following recovery key 'gvfphxiasuva'

Key: gvfphxiasuva

First Contact
Score: 40

> You notice that the indicator light near the robot's antenna begins to blink. Perhaps the robot is connecting to a network? Using a wireless card and the network protocol analyzer Wireshark, you are able to create a PCAP file containing the packets sent over the network.
>
> You suspect that the robot is communicating with the crashed ship. Your goal is to find the location of the ship by inspecting the network traffic.

-패킷들을 보다보면 중간부터 해시값에 문자열이 보이기 시작합니다.

> "..ROBOT BOOT UP INITIALIZING..
> ..SPACE SHIP READY..
> […중략]
> ..NEW LOCATION COORDINATES: '06"N '40"W.."

-'06"N '40"W를 구글링하면 Area 51이라고 나옵니다.

Key: Area 51

Spaceport Map
Score: 55

> The map of the spaceport is hard to parse, but you're pretty sure there is some hidden information, somewhere…

-사진들이 휙휙 지나갑니다. 오른쪽 아래에 보면 밑에 있는 문자열이 자꾸 바뀌는 것을 볼 수 있습니다. 자세히 보다 보면, 중간에 Key뭐시기가 보이네요. 계속 반복하여 보거나, 저장하여 한장한장 봅시다.

Key: Do passports let you fly interstellar?

GETKey
Score: 50

> There's bound to be a key on the spaceport's hidden website
> https://picoctf.com/problems/getquery/index.php?admin=false&competition=ccdc

-우선 Get Key! 버튼이 있으니 눌러봅니다.
index.php?admin=false&competition=ccdc 이 부분이 마음에 걸립니다. 소스를 봅시다.

```
<input type="hidden" name="admin" value="false" />
<input type="hidden" name="competition" value="ccdc" />
```

-name="admin" value="true"와 name="competition" value="picoctf" 로 수정해 준 후, 새로고침 하면..
FLAG: 9fa449c061d64f58de600dfacaa6bd5d 가 뜹니다.

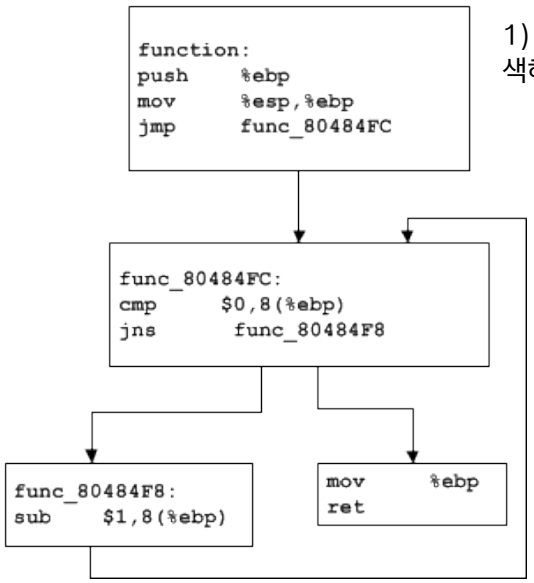
Key: 9fa449c061d64f58de600dfacaa6bd5d

CFG to C
Score: 70

Wouldn't it be cool to be able to have one of these patrol drones to do your bidding?! Figure out the correct sequence of C functions from the following control flow graphs and you should be well on your way.

Submit the correct order of functions.

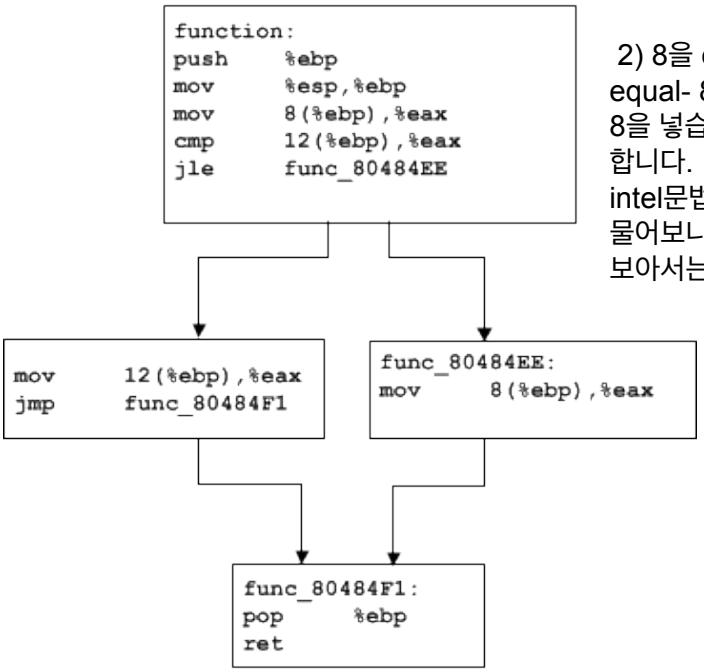
-아.. 어셈 잘 몬하는데..



1) func_80484FC로 간 후 0과 8,(ebp)를 서로 비교 한 후 jns를 하는데 검색해보니 jns는 Jump if sign flag not set랍니다.
Sign flag-이것은 연산 결과 최상위 비트의 값과 같다. Signed 변수의 경우 양수이면 0, 음수이면 1이 된다.
솔직히 말하자면 뭔소린지 모르겠는데 아마도 cmp한 결과가 1이 아니면(0과 ebp가 같지 않으면) func_8048F8로 가고 그 후 func_8048F8에서 ebp에서 하나를 빼고 다시 돌려보냅니다. ebp가 0이면 프로그램을 끝냅니다. 전형적인 루프문입니다.

B)

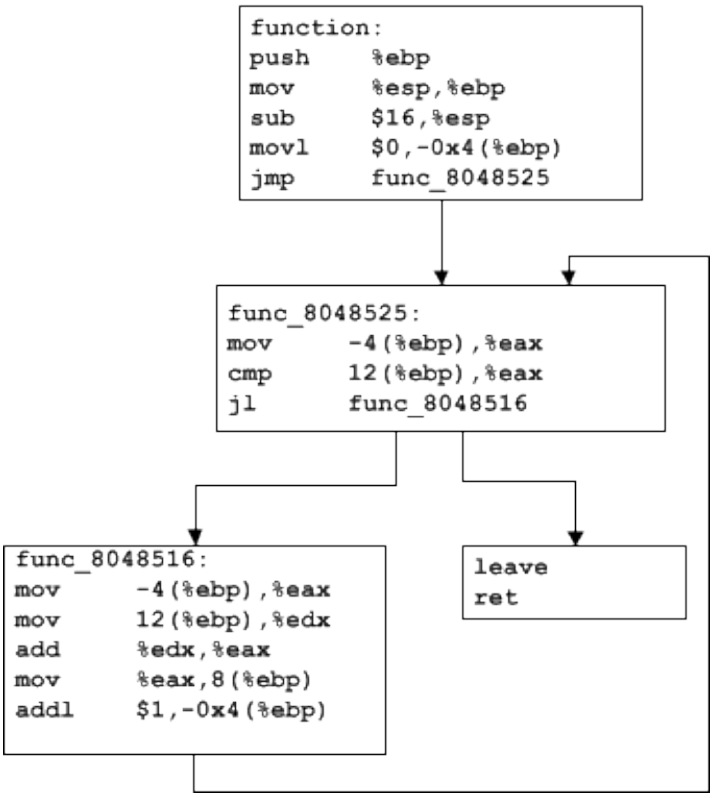
```
int loop(int a)
    {
      while (a >= 0) {
        a--;
      }
      return a;
    }
```



2) 8을 eax에 넣고 12랑 8을 비교합니다. jle= jump if less or equal- 8이 12보다 작거나 같으면 func_80484EE로 넘어가 eax에 8을 넣습니다. 아니라면 eax에 12를 넣어주고요. 그런 후 ebp를 리턴합니다.
intel문법이라면 mov가 인자 순서가 달라서 헷갈렸는데 팝콘 형님께 물어보니 프롤로그를 보면 intel은 mov ebp, esp일텐데 그 반대인걸 보아서는 at&t라고 말씀해주셨습니다.
…천재다.
암튼 고로 이 어셈에 해당하는 c 코드는:

C)

```
int control(int a, int b)
{
  if (a > b)
    return b;
  else
    return a;
}
```



3) 음. 이번엔 좀 괴상망측하네요.
그래도 뭐 movl과 addl은 같은 mov와 add를 long에 적용시킨것이니까 같게 생각하면 되겠죠. 우선 esp를 16만큼 빼네여. 스택에 16만큼의 공간을 만들어주는거죠. 그런 후 ebp에 0을 넣고, func_8048525로 뛉니다. func_8048525에선 eax에 -4를 넣은 후, 12와 eax(-4)를 비교한 후 eax가 더 크면 끝나고 더 작으면 func_8048516으로 갑니다. 여기선 eax에 -4를 넣고 edx에 12를 넣고 둘을 더한 후에 ebp에 계산결과를 넣고 ebp에 1을 더하네여. 그런 후 돌아갑니다.

D)

```
int for_loop(int a, int b, int c)
    {
        int i;

        for(i = 0; i < b; i++)
            a = c + i;
        return a;
    }
```

```
function:
push   %ebp
mov    %esp,%ebp
mov    12(%ebp),%eax
mov    %eax,%edx
sar    $0x1f,%edx
idiv   8(%ebp)
mov    %edx,%eax
pop    %ebp
ret
```

4) 여긴 하나의 함수밖에 보이지 않네요. eax에 12를 넣고 edx에도 넣은 뒤, sar가 오른쪽으로 1f만큼 쉬프트 연산을 하는거라네여. 그리고 idiv는 12를 8로 나눈 후 그 몫을 eax에 넣고 나머지를 edx에 넣습니다.

A)

```
int modulo(int a, int b)
    {
        return b % a;
    }
```

Key: BCDA

Bitwise
Score: 55

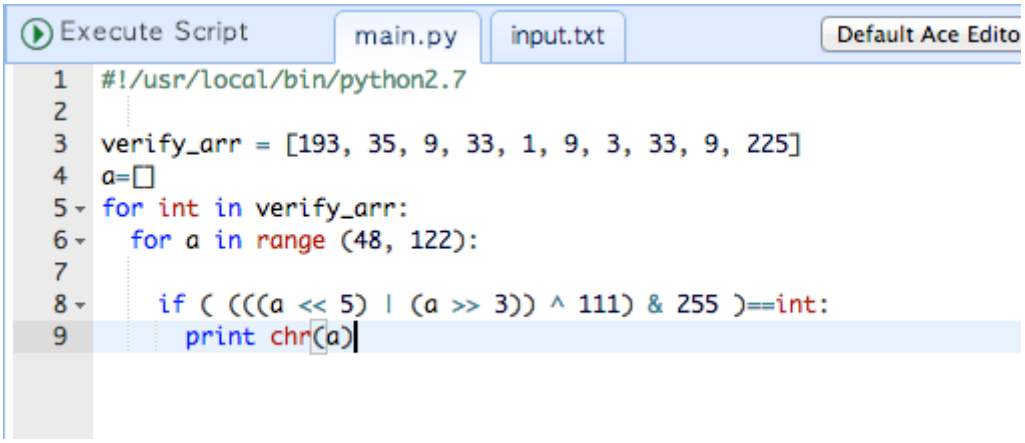
You see the doors to the loading bay of the hangar, but they are locked. However, you are able to extract the password verification program from the control panel... Can you find the password to gain access to the loading bay?

bit.py:

```
#!/usr/bin/env python

user_submitted = raw_input("Enter Password: ")

if len(user_submitted) != 10:
  print "Wrong"
  exit()
```

```
verify_arr = [193, 35, 9, 33, 1, 9, 3, 33, 9, 225]
user_arr = []
for char in user_submitted:
  # '<<' is left bit shift
  # '>>' is right bit shift
  # '|' is bit-wise or
  # '^' is bit-wise xor
  # '&' is bit-wise and
  user_arr.append( (((ord(char) << 5) | (ord(char) >> 3)) ^ 111) & 255 )

if (user_arr == verify_arr):
  print "Success"
else:
  print “Wrong”
```

-비트연산으로 입력받은 값을 바꾼 후에 위의 숫자 배열과 비교해보는데요, 역산하는 방법도 있지만 다른 방법을 사용했습니다. 파이썬 잘 못해서 처음에는 막막했는데 잘 되드라구여.



-이 스크립트를 실행시키니 키가 나왔습니다. 이 스크립트는 아스키값에서 숫자부터 영어 대문자까지 각각 계산을 한 뒤 위의 verify_arr과 비교하고 값이 맞으면 문자로 출력해줍니다.

Key: ub3rs3cr3t

Yummy
Score: 60

You want to find out the docking bay numbers for space ships that are ready to launch. Luckily for you, the website for the docking bay ship status page doesn't seem so secure....
Enter the docking bay for any of the ships that are awaiting launch.

-개발자 도구를 통해 소스를 조금 들여다보면..

<!-- DEBUG: Expected Cookie: “authorization=administrator"
received Cookie: "" —>

이런 주석을 볼 수 있습니다. (구글 크롬 익스텐션인 Edit This Cookie!를 사용하였슴다) 이걸 이용해 authorization 쿠키를 만들고 안에 administrator란 값을 넣은 후 새로고침하면..

Docking Bay Ship Status

Login Success.

Docking Bay	Ship Designation	Status
DX9-2	Proud Planetary Prowler	Under Repair
DX5-2	Century Eagle	Fuelling
DX6-7	USS Franchise	Awaiting Launch
DX7-1	HMS Beagle	Under Repair
DX7-2	Pwnie Express	Awaiting Launch
DX9-5	Happy Fun Time Awesome Best Ship	Awaiting Launch
DX2-2	Roflcopter	Under Repair
DX4-5	YHXS-2 Ghost	Under Repair
DX4-9	Theseus	Awaiting Launch
DX8-2	Drone DK97	Fuelling

-Login Success란 메세지와 함께 Ship들의 상태가 나옵니다. 이 중 Awaiting Launch 상태인 우주선들 중 하나의 Docking Bay를 골라 넣습니다.

Key: DX6-7 || DX7-2 || DX9-5 || DX4-9

Injection
Score:110

Use the Loading Bay Control System to get the admin key! Problem

Hint: The SQL query being performed is something a lot like SELECT username,hash FROM pwtable WHERE username= your_input

-제가 웹은 잘 모릅니다. 하지만 Username 을 넣는 곳에 '=' 를 치고 엔터를 누르게 되면 유저네임들과 함께 해쉬가 나옵니다. 그 중 맨 마지막에 있는 어드민의 해쉬가 키입니다.

Key: bad_code_and_databases_is_no_fun

Pilot Logic
Score:75

You've gotten a partial dump of the disk from the hangar's machine, and you're pretty sure the pilot's password is cleverly hidden somewhere within it...

The disk image can be found on the shell machines at /problems/pilot_logic.img and the contents of the image are available in /problems/pilot_logic/

-직접 picoctf 서버로 들어갔습니다. 처음에 ls 했는데도 home 이랑 lost+found밖에 안보여 연락드린 PPP분들께 죄송하다는 말씀 드립니다.. 으헝헝. 한분도 아니고.. 정말 죄송합니다ㅠㅠ 암튼 처음 접속하면..

```
$ cd /problems/pilot_logic
$ ls
home  lost+found
$ cd home
$ ls
bureaubot  guardbot  janitorbot  pilotbot
```

-이런 디렉토리들이 보입니다. 각각의 디렉토리에 들어가 ls -al을 하다보면..

```
$ cd pilotbot
$ ls -al
total 8
drwxrwxr-x 8 root root 1024 Mar 24  2013 .
drwxrwxr-x 6 root root 1024 Mar 24  2013 ..
drwxrwxr-x 2 root root 1024 Mar 24  2013 Desktop
drwxrwxr-x 2 root root 1024 Mar 24  2013 Documents
drwxrwxr-x 2 root root 1024 Mar 24  2013 Music
drwxrwxr-x 2 root root 1024 Mar 24  2013 Pictures
drwxrwxr-x 2 root root 1024 Mar 24  2013 Public
drwxrwxr-x 2 root root 1024 Mar 24  2013 .secret
$ cd .secret
$ ls
key
$ cat key
"You can't take the sky from me"
```

Key: You can't take the sky from me