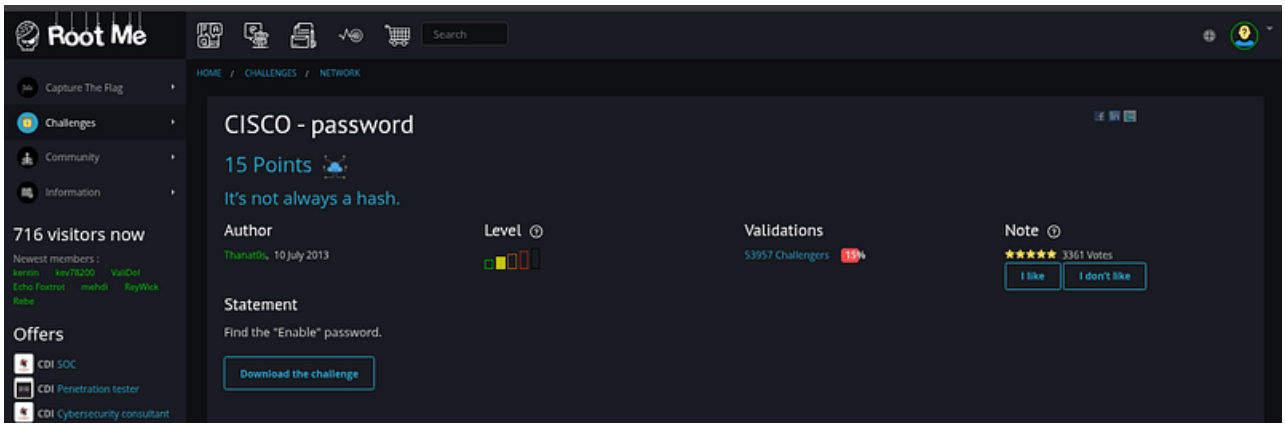
Root-Me Write-up: Cisco Password



Tools Used

- Cisco Type 7 decoder: https://github.com/theevilbit/ciscot7

Initial Hint

The challenge hint was **“Cisco password”**, which strongly suggests Cisco-specific password storage formats rather than standard hashes.

After a quick search, I found that Cisco commonly uses multiple password types. A useful reference was:

- https://www.firewall.cx/cisco/cisco-routers/cisco-type7-password-crack.html

This confirmed that **Type 7 passwords are reversible**, while others (such as Type 5) are not.

Analysis

While reviewing the configuration, I noticed the presence of:

enable secret 5 ...

Research showed that **Type 5 (MD5-crypt) passwords cannot be decrypted**, only cracked. Since the challenge did not require brute-forcing, this indicated that the enable secret 5 entry was likely a distraction.

Encryption Methods That Cannot Be Decrypted

As opposed to **Type 7 Passwords** which can easily be decrypted, **Secret 5 passwords cannot be decrypted** as the password has ben hashed with **MD5**. This is also the recommened way of creating and storing passwords on your Cisco devices.

Following are a number of examples where **Secret 5** passwords can and should be used:

USER PASSWORDS

```
# username chris privilege 15 secret 5 $1$KNaN$SCe/xMbtBEe6ch5d2bq5J.
```

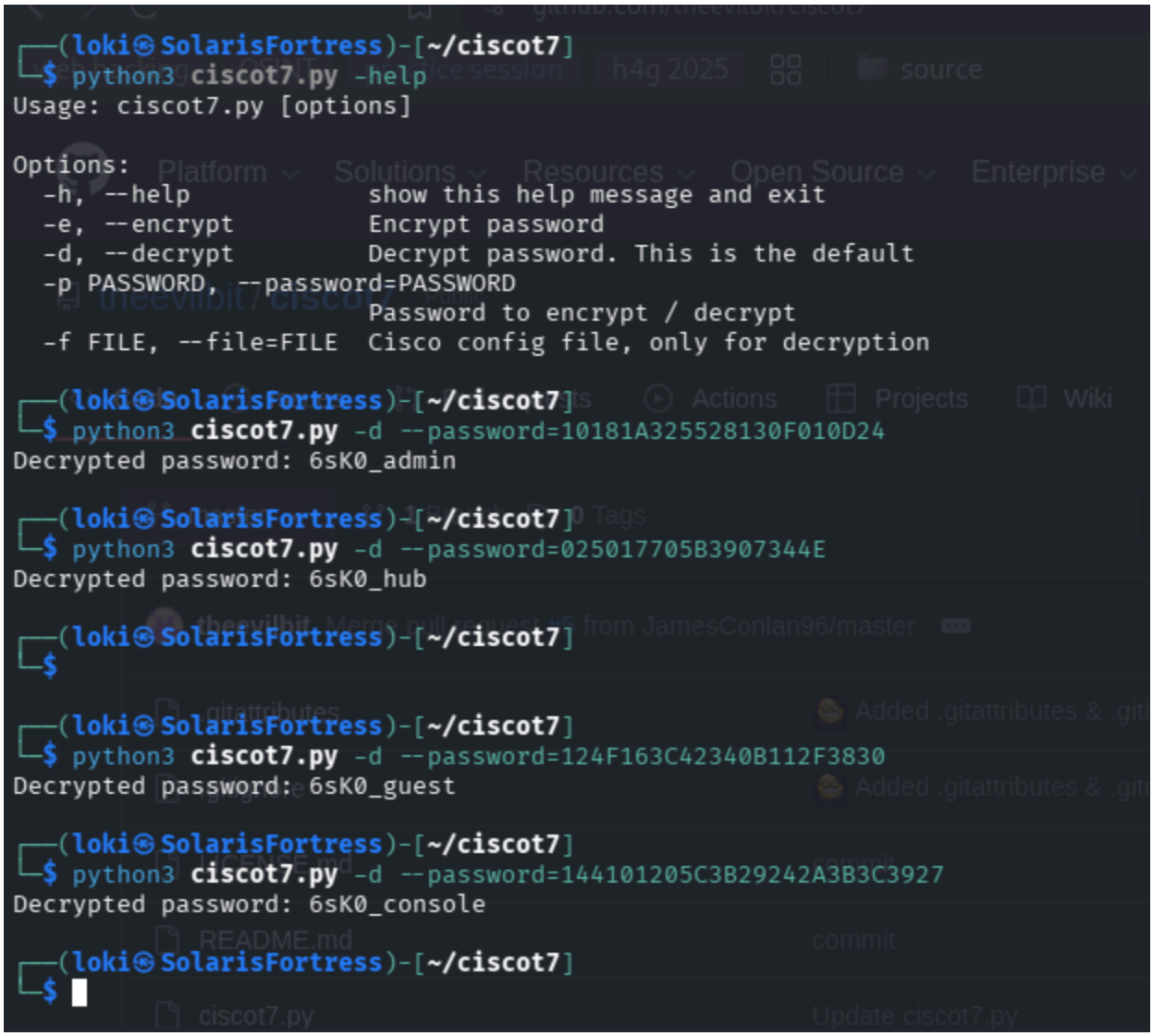
ENABLE PASSWORD

```
# enable secret 5 $1$2UjJ$cDZ05dfEGA7mHfE4RSbWiQ.
```

The configuration also contained **multiple password 7 entries**, which *are* reversible. This aligned perfectly with the challenge hint.

Decoding Cisco Type 7 Passwords

I extracted all Type 7 passwords from the configuration and decoded them using ciscot7.py.



Encoded Passwords

username hub password 7 025017705B3907344E
username admin privilege 15 password 7 10181A325528130F010D24
username guest password 7 124F163C42340B112F3830
password 7 144101205C3B29242A3B3C3927

Decoded Results

10181A325528130F010D24 → 6sK0_admin
025017705B3907344E → 6sK0_hub
124F163C42340B112F3830 → 6sK0_guest
144101205C3B29242A3B3C3927 → 6sK0_console

At this point, a clear pattern emerged: **the string 6sK0 was consistent**, with different suffixes indicating the context or role of each password.

Identifying the Flag

Cisco devices have a special **privileged mode** called **enable**. Accessing this mode is a key escalation step and is commonly the objective in Cisco-based CTF challenges.

Given:

- The consistent base string 6sKo
- The role-based suffix pattern (_admin, _hub, _guest, _console)
- The importance of privileged (enable) mode

It was reasonable to conclude that the enable password follows the same convention:

6sKo_enable

Final Flag

6sKo_enable

Configuration Snippet

```
!
! Last configuration change at 13:41:43 CET Mon Jul 8 2013 by admin
! NVRAM config last updated at 11:15:05 CET Thu Jun 13 2013 by admin
!
version 12.2
no service pad
service password-encryption
!
hostname rmt-paris
!
```

```
security passwords min-length 8
no logging console
enable secret 5 $1$p8Y6$MCdRLBzuGlfOs9S.hXOp0.
!
username hub password 7 025017705B3907344E
username admin privilege 15 password 7 10181A325528130F010D24
username guest password 7 124F163C42340B112F3830
!
line con 0
 password 7 144101205C3B29242A3B3C3927
 session-timeout 600
line vty 0 4
 session-timeout 600
 authorization exec SSH
 transport input ssh
```

By Alexander Sapo on December 16, 2025.

Canonical link

Exported from Medium on February 7, 2026.