|  |  |
| --- | --- |
| **팀 이름**  **Team Name** | EHC\_fumosquad |
| **문제 이름**  **Question** | *Invalid OTP* |
| 문제 풀이과정 작성 (캡처화면 필수) / Write-up Details (The screenshot is mandatory) | |
| Link: <http://hacktheon2025-challs-alb-1354048441.ap-northeast-2.elb.amazonaws.com:39406/>  After enter the link, in f12, we can see some image have source like this: <img style="max-width:80%" src="/image\_render?url=https://cdn.jsdelivr.net/gh/ott3r07/nexora@51d875a2dcc807005a9f9f416c1636bf686e7405/logo.png"/>  Endpoint /image\_render is vulnerable by ssrf. How about localhost???    It is filtered, but I see something familiar ….: <https://www.leviathansecurity.com/blog/bypassing-ssrf-filters-using-r3dir>  Using this method, we can bypass this filter.  At first, we access this link:  http://hacktheon2025-challs-alb-1354048441.ap-northeast-2.elb.amazonaws.com:39406/image\_render?url=https%3A%2F%2F307.r3dir.me%2F--to%2F%3Furl%3Dhttp%3A%2F%2Flocalhost    Nice, now we can see the source and all seed of all session.    Here is the credential.  Login and phase 2 is comingggg, bypass otp  After read source, it is a python program with a otp random with seed in otp\_seed.json. If you guess right otp, then flag.    /otp\_gen endpoint will print the result of random. Now, we need to guess the right otp. Observe the otp\_gen endpoint, i think it is not different from random with seed, so I write a code to predict next otp, and it true (I will show code later). But, when I test it in web, it isn’t true. After some try, I realize that in endpoint /otp, endpoint /otp\_gen is accessed 2 time. Therefore, I must take the result following the next result. Here is the code to gen otp  “””  import random  seed = 1745647185 # go to otp\_seed.json to get it  rng = random.Random(seed)  otp1 = 1294646144 # go to otp\_gen?session\_id to get it  for i in range(100):  otp2 = rng.getrandbits(32)  if( otp2 == otp1):  otp3= rng.getrandbits(32)  otp4= rng.getrandbits(32)    print(otp4) # the true otp you enter  “””  After enter this code in /otp endpoint, you will have flag    FLAG: FLAG{55rf\_byp455\_0tp\_prn9\_345y\_cr4ck3d} | |