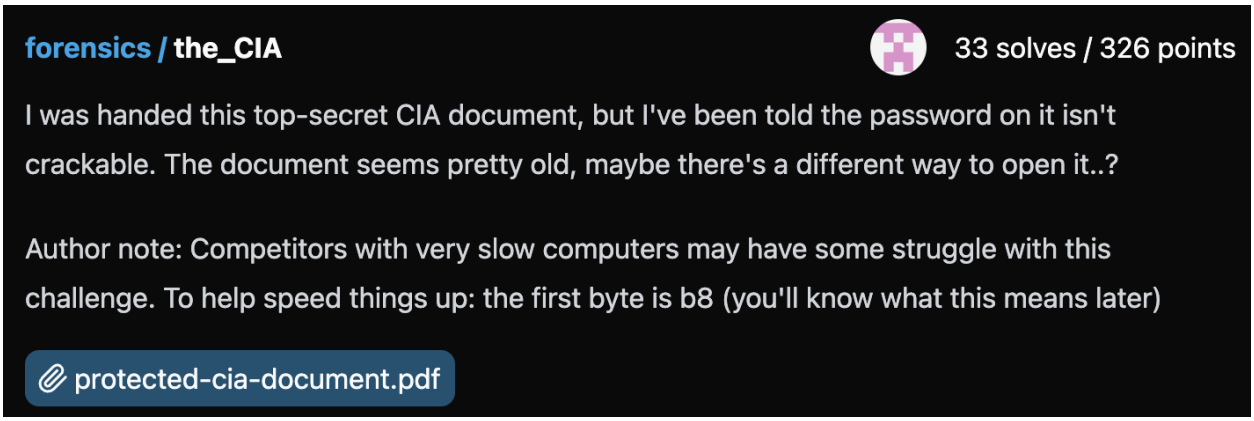
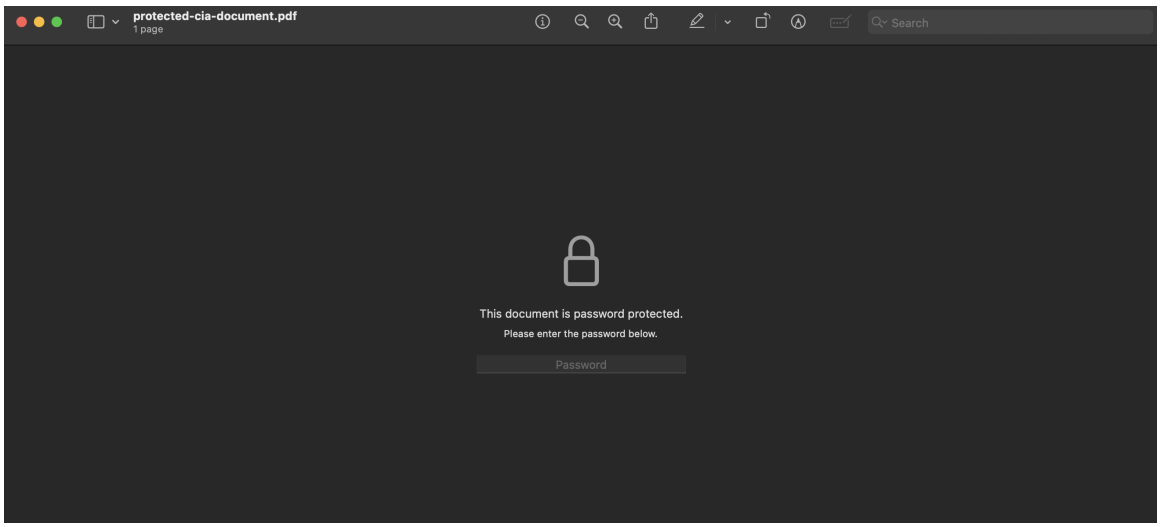
the_CIA

Challenge



Introduction

문제에서 주어진 `.pdf` 파일을 먼저 열어봤습니다.



문제에서 말한대로 암호화가 걸려있다고하여 내부 정보는 확인할 수 없음

exiftool을 활용하여 해당 파일의 세부 정보를 확인함

```
┌──(parallels㉿kali-gnu-linux-2023)-[~/Desktop]
└─$ exiftool protected-cia-document.pdf
ExifTool Version Number         : 12.57
File Name                       : protected-cia-document.pdf
Directory                       : .
File Size                       : 117 kB
File Modification Date/Time     : 2024:10:03 16:23:40+09:00
File Access Date/Time           : 2024:10:07 10:36:19+09:00
File Inode Change Date/Time     : 2024:10:07 10:22:02+09:00
File Permissions                : -rw-r--r--
File Type                       : PDF
File Type Extension             : pdf
MIME Type                       : application/pdf
PDF Version                     : 1.7
Linearized                      : No
Encryption                      : Standard V1.2 (40-bit)
User Access                     : Print, Modify, Copy, Annotate, Fill forms, Extract, Assemble, Print high-res
Warning                         : Document is password protected (use Password option)
```

Standard V1.2로 암호화된 상태이며, 40비트의 크기로 암호화가 진행되어 표준 등급의 보안이 유지되고 있는 것을 확인함

따라서, 해당 비밀번호를 알아내어 pdf 내용물을 보는 것을 목표로 함

> Author note: Competitors with very slow computers may have some struggle with this challenge. To help speed things up: the first byte is b8 (you'll know what this means later)

문제에서 비밀번호의 첫 바이트가 b8로 시작한다는 힌트를 주었으니 Brute Force로 40바이트 비밀번호를 알아내는 방향으로 접근함

`pdf2john` 이라는 암호 해시 크래킹 도구를 사용하여 PDF 파일의 암호화된 데이터를 **해시 값**으로 변환하여 처리할 수 있도록 함

> 💡 pdf1240*-4116*f1707cb82f3dbf48b43ba62b159dd92f32ec946c5b13a86b1e83ace77cae236219520f343c318080a

`$pdf$1 2 40*-4 1 16`

이 부분은 pdf 포멧을 보여주고, 1.2 표준 암호화로 40비트 비밀번호임을 나타냄

`f1707cb82f3dbf48b43ba62b159dd92f` `32` `ec946c5b13a86b1e83ace77cae236219520f343c318080a08b5032fed386c369` `32*3e6bcb942137ae9c4d3530581158fc277eeb`

뒤의 이 부분은 pdf 파일의 암호화된 데이터 해시 값임을 알 수 있습니다.

이 파트를 `hash.hash` 파일로 하여 저장합니다.

```
┌──(parallels㉿kali-gnu-linux-2023)-[~/Desktop]
└─$ cat hash.hash
$pdf$1*2*40*-4*1*16*f1707cb82f3dbf48b43ba62b159dd92f*32*ec946c5b13a86b1e83ace77cae236219520f343c318080a08b5032fed386c369*32*3e6bcb942137ae9c4d3530581158fc277eeb794952f3ceaa76389183fa
5dda55
```

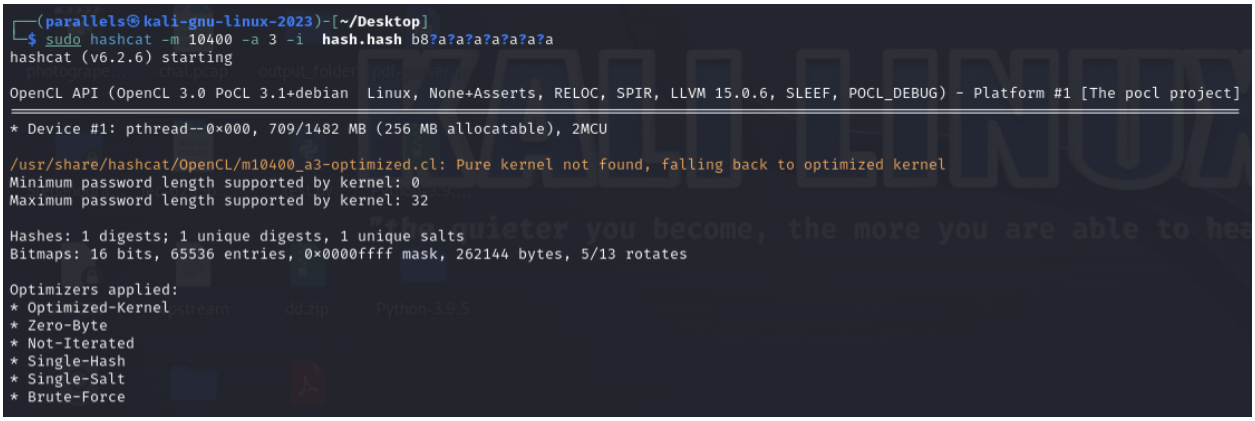
`hashcat` 이라는 해시값 크래킹 비밀번호 복구 도구를 사용하여 비밀번호 복호화를 진행함

다음과 같은 명령어를 통해 진행함

> 💡 sudo hashcat -m 10400 -a 3 -i hash.hash b8?a?a?a?a?a?a?a

- `m 10400` : 이 옵션은 해시 유형을 지정합니다. 여기서 `10400`은 PDF 파일에서 사용된 암호화 방식(PDF 1.4 - 1.6)을 의미합니다.
- `a 3` : 공격 방식을 지정하는 옵션입니다. `3`은 **브루트포스 공격**을 의미합니다. 즉, 가능한 모든 문자 조합을 통해 비밀번호를 추측하는 방식입니다.

- `i`: 이 옵션은 **점진적 크래킹(incremental)**을 의미합니다. 즉, 비밀번호 길이가 짧은 것부터 긴 것까지 순차적으로 탐색합니다.
- `hash.hash`: 크래킹할 해시가 저장된 파일입니다. 이 파일에는 PDF 파일의 암호화된 해시가 담겨 있습니다.
- `b8?a?a?a?a?a?a`: 이것은 **비밀번호 마스크**를 지정하는 부분입니다. 여기서 `b8`로 시작하고 나머지 6자리는 임의의 문자(`?a`)로 구성된 8자리 비밀번호를 탐색하겠다는 의미입니다. `?a`는 모든 가능성(대문자, 소문자, 숫자, 특수문자)을 포함한 임의의 문자를 나타냅니다.



실행결과는 다음과 같습니다.

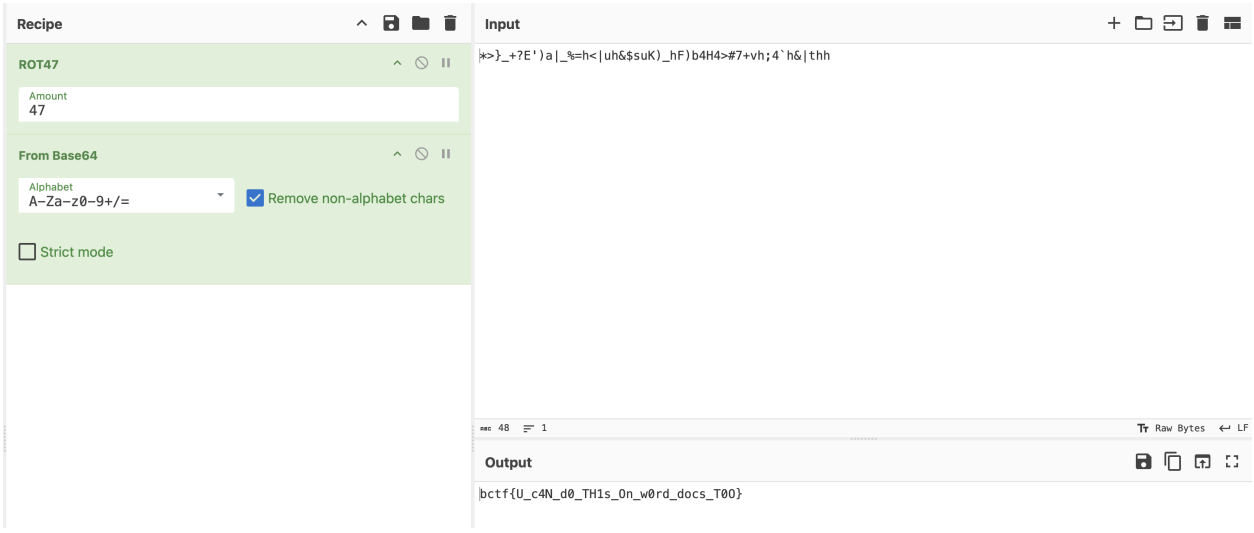
> pdf1240*-4116f1707cb82f3dbf48b43ba62b159dd92f32ec946c5b13a86b1e83ace77cae236219520f343c318080a
> 42137ae9c4d3530581158fc277eeb794952f3ceaa76389183fa5dda55:b8?8uj[{1
>
> Session.....: hashcat
> Status.....: Cracked
> Hash.Mode.....: 10400 (PDF 1.1 - 1.3 (Acrobat 2 - 4))
> Hash.Target.....: pdf1
> 240*-4116*f1707cb82f3dbf48b43ba62b159dd92...5dda55
> Time.Started.....: Sat Sep 28 17:26:14 2024 (16 mins, 24 secs)
> Time.Estimated...: Sat Sep 28 17:42:38 2024 (0 secs)
> Kernel.Feature...: Optimized Kernel
> Guess.Mask.....: b8?a?a?a?a?a?a [9]
> Guess.Queue.....: 9/18 (50.00%)
> Speed.#1.....: 1502.3 MH/s (11.62ms) @ Accel:256 Loops:32 Thr:32 Vec:1
> Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
> Progress.....: 1450859036672/69833729609375 (2.08%)
> Rejected.....: 0/1450859036672 (0.00%)
> Restore.Point....: 160432128/7737809375 (2.07%)
> Restore.Sub.#1...: Salt:0 Amplifier:5280-5312 Iteration:0-32
> Candidate.Engine.: Device Generator
> Candidates.#1....: b8@Kkqe{1 → b8\A}}W||
> Hardware.Mon.#1..: Temp: 69c Fan: 66% Util: 98% Core:1965MHz Mem:9251MHz Bus:16

pdf 파일 비밀번호는 `b8?8uj[{1` 임을 알 수 있음

이를 pdf 파일에 넣으면 열리는 것을 알 수 있다.

내용을 보면, ROT47로 암호화, Base64로 암호화를 진행한 것을 알 수 있음

Cyberchef 사이트에서 복호화를 진행하여 Flag 값을 확인할 수 있음



`bctf{U_c4N_d0_TH1s_On_w0rd_docs_T0O}`