Weak Password

CSAW 2021 CTF

OreoByte

Weak Password

50

Can you crack Aaron's password hash? He seems to like simple passwords. I'm sure he'll use his name and birthday in it. Hint: Aaron writes important dates as YYYYMMDD rather than YYYY-MM-DD or any other special character separator. Once you crack the password, prepend it with `flag{` and append it with `}` to submit the flag with our standard format. Hash: `7f4986da7d7b52fa81f98278e6ec9dcb`.

Author: `moat`, Pacific Northwest National Laboratory

2021-09-10

Contents

Weak Password

Writeup by: OreoByte

Team: OnlyFeet

Writeup URL: GitHub

```
Can you crack 'Aarons password hash? He seems to like simple passwords.
    'Im sure 'hell use his name and birthday in it. Hint: Aaron writes
   important dates as YYYYMMDD rather than YYYY-MM-DD or any other
   special character separator. Once you crack the password, prepend it
    with flag{ and append it with } to submit the flag with our
   standard format. Hash: 7f4986da7d7b52fa81f98278e6ec9dcb.
```

Scripting it Out with and Bash

First we create the dates part of the wordlist with a script. Then use some bash and sed-foo to further modify the dates file `mdy` with name and special characters.

```
#!/usr/bin/python3
# month/day/year
from datetime import date, timedelta
sdate = date(1900,1,1)
edate = date(2022,1,1)
delta = edate -sdate

for i in range(delta.days + 1):
  day = sdate + timedelta(days=i)
  #print(day.month,day.day,day.year)
  print(''.join([str(day.year), str(day.month), str(day.day)]))
```

```
python solve.py > mdy

#!/bin/bash
sed -e 's/^/Aaron/' mdy > cap_user_date
sed -e 's/^/aaron/' mdy >> cap_user_date
sed -e 's/$/aaron/' mdy >> cap_user_date
sed -e 's/$/Aaron/' mdy >> cap_user_date
sed -e 's/^/Aaron /' mdy >> cap_user_date
sed -e 's/$/Aaron /' mdy >> cap_user_date

# special chars
cp cap_user_date final.lst
```

```
sed -e 's/$/!/' cap_user_date >> final.lst
sed -e 's/$/@/' cap_user_date >> final.lst
sed -e 's/$/#/' cap_user_date >> final.lst
sed -e 's/$/$/' cap_user_date >> final.lst
sed -e 's/$/%/' cap_user_date >> final.lst
sed -e 's/$/^/' cap_user_date >> final.lst
sed -e 's/$/&/' cap_user_date >> final.lst
sed -e 's/$/*/' cap_user_date >> final.lst
for i in {0..9}; do sed -e 's/$/$i/' cap_user_date >> final.lst; done
```

Let's Get Crackin'

User hashid to identify the hash and then we can crack it with hashcat and the custom wordlist we just previously generated and a Hashcat rule to expand the wordlist a bit.

`hashid -m 7f4986da7d7b52fa81f98278e6ec9dcb` OR Hash Analyzer

```
$ hashcat 7f4986da7d7b52fa81f98278e6ec9dcb final.lst -r
   OneRuleToRuleThemAll.rule
$ hashcat 7f4986da7d7b52fa81f98278e6ec9dcb final.lst -r
   OneRuleToRuleThemAll.rule --show
7f4986da7d7b52fa81f98278e6ec9dcb:Aaron19800321
```

Victory

Submit the flag and claim the points:

flag{Aaron19800321}