# #

CTF 01/04 - Crack the Hash

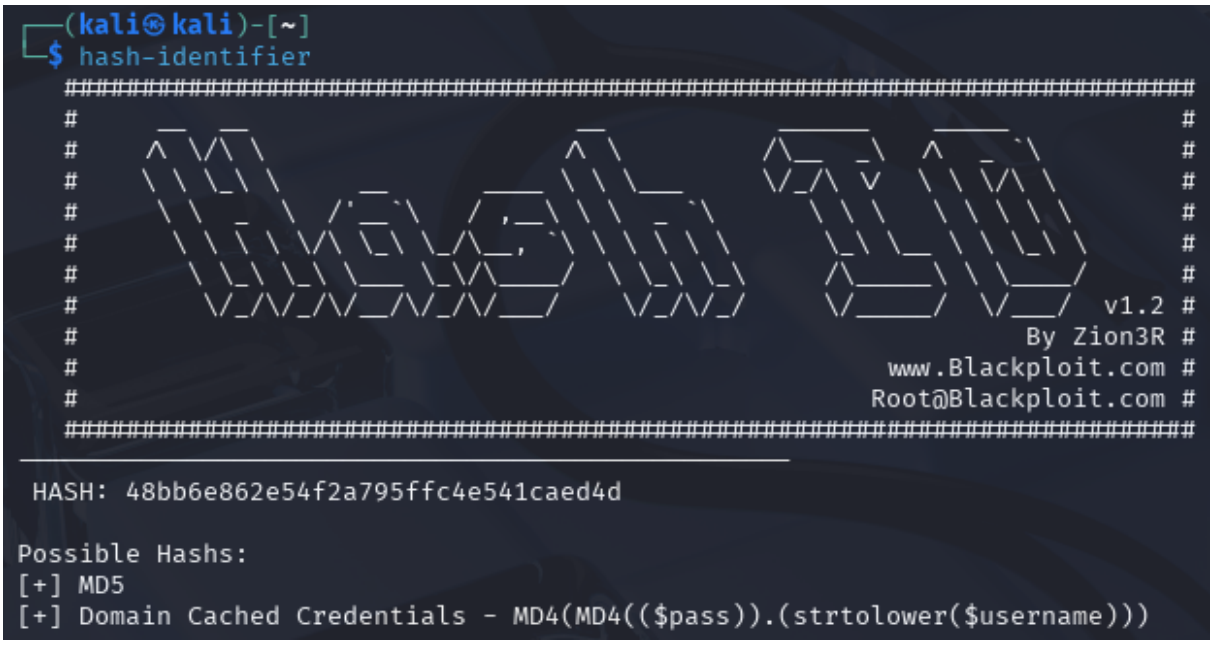
Pra realização desse CTF serão necessárias basicamente duas ferramentas:

1. hashid ou hash-identifier
2. hashcat

No nível 1, tudo pode ser realizado inteiramente em sites próprios para a resolução de hashs, mas como a ideia da UFSC - OFFSEC é ser um grupo de segurança ofensiva eu priorizei o uso do kali sempre que possível.

NÍVEL 1

A primeira hash que temos é `48bb6e862e54f2a795ffc4e541caed4d` e para identificar qual função foi utilizada na sua produção, utilizamos o hash-identifier:



Infelizmente, nem sempre o resultado das possíveis hashs é, de fato, a que foi utilizada mas muitas vezes ela já dá uma ideia.

Agora, tendo a função (MD5) utilizei `hashcat -a 0 -m 0 48bb6e862e54f2a795ffc4e541caed4d /usr/share/wordlists/rockyou.txt` onde:

- `-a` é o modelo de ataque, sendo 0 um ataque de dicionário
- `-m` é a identificação da função de hash, sendo 0 referente à MD5
- `48bb6e862e54f2a795ffc4e541caed4d` é a hash a ser quebrada
- `/usr/share/wordlists/rockyou.txt` é o caminho para a wordlist que eu utilizei no ataque

Após alguns segundos, tive o output `48bb6e862e54f2a795ffc4e541caed4d:easy` , o que sinaliza que a primeira flag é **"easy".**

Para as seguintes hashs o procedimento segue o mesmo padrão, com algumas diferenças...

O hash-identifier nos diz que `CBFDAC6008F9CAB4083784CBD1874F76618D2A97` é SHA-1, que tem código `100` no hashcat. Esse código pode ser verificado tanto com o comando `hashcat -h` quanto conferindo esse site.

O comando utilizado foi `hashcat -a 0 -m 100 CBFDAC6008F9CAB4083784CBD1874F76618D2A97 /usr/share/wordlists/rockyou.txt` e temos como resposta a flag **"password123".**

Em seguida, novamente o hash-identifier sugere que `1C8BFE8F801D79745C4631D09FFF36C82AA37FC4CCE4FC946683D7B336B63032` é SHA-256, código `1400` .

Utilizando `hashcat -a 0 -m 1400 1C8BFE8F801D79745C4631D09FFF36C82AA37FC4CCE4FC946683D7B336B63032 /usr/share/wordlists/rockyou.txt` temos como resposta a flag **"letmein".**

Na sequência, o hash-identifier não deve conseguir identificar `$2y$12$Dwt1BZj6pcyc3Dy1FWZ5ieeUznr71EeNkJkUlypTsgbX1H68wsRom` propriamente, mas podemos analisar a lista de exemplos de hash e identificar que o padrão inicial com `$2` é referente à `bcrypt $2*$, Blowfish (Unix)` , que tem código igual a `3200` .

Nesse caso, por algum motivo que eu desconheço, não é possível utilizar a hash diretamente na linha de comando, então eu criei um arquivo hash.txt e coloquei a mesma lá dentro.

Uma otimização que pode ser feita nesse caso, senão será um processo bem demorado, é transformar a nossa wordlist em outra um pouco menos, visto que

é possível de observar que a resposta no TryHackMe contém apenas 4 letras. Utilizei `grep -E '^[a-z]{4}$' /usr/share/wordlists/rockyou.txt > list.txt` onde:

- `grep` é um comando usado pra encontrar padrões em textos
- `^` limita o início da linha
- `-E` é utilizado para ativar a sintaxe de expressões regulares estendidas
- `[a-z]` limita os caracteres à letras minúsculas
- `{4}` limita o número de caracteres à 4
- `$` limita o fim da linha
- `>` faz com que o resultado do `grep` seja direcionado para `list.txt`

Então, finalmente eu utilizei `hashcat -a 0 -m 3200 hash.txt list.txt` e obtive a resposta **"bleh".**

No último desafio desse nível, além de o hash-identifier não sugerir MD4 como uma das maiores possibilidades (e é o tipo dela, de acordo com a dica do desafio) eu não consegui utilizar o hashcat pra quebrar a hash de jeito nenhum, sempre tive o `status: Exhausted` então utilizei um decriptador online qualquer de hash MD4. O retorno desses foi a resposta **"Eternity22".**

E finalmente... Fim do nível 1 🥳🥳🥳

NÍVEL 2

À partir desse ponto que o desafio esperava que começássemos a utilizar o hashcar, mas como a gente é muito bom já estamos adiantados. Então, as resoluções seguem os passos do nível anterior, com pequenas alterações...

Primeiramente, o hash-identifier nos diz que `F09EDCB1FCEFC6DFB23DC3505A882655FF77375ED8AA2D1C13F640FCCC2D0C85` é do tipo SHA-256, que a gente já utilizou anteriormente. Então, utilizando `hashcat -a 0 -m 1400 F09EDCB1FCEFC6DFB23DC3505A882655FF77375ED8AA2D1C13F640FCCC2D0C85 /usr/share/wordlists/rockyou.txt` obtemos a resposta **"paule".**

A seguinte tem aquele mesmo problemas no hash-identifier onde ele nos diz que se trata de um MD5 quando na verdade é NTLM (como a gente vê na dica),

que tem o código `1000`.

Sabendo disso, utilizando `hashcat -a 0 -m 1000 1DFECA0C002AE40B8619ECF94819CC1B /usr/share/wordlists/rockyou.txt` obtemos a flag **“n63umy8lkf4i”.**

A seguinte foi um pouco mais complicada de fazer, mas vai ser mais tranquilo de explicar já sabendo da resposta...

Novamente o hash-identifier não consegue identificar (irônico, né?) o tipo de hash, então fui analisar no site com exemplos de hash aquelas que começam com `$6` e logo a primeira que aparece é sha512crypt 6, SHA512 (Unix), com código `1800`. É importante perceber que nessa função o salt já vai integrado na hash, sem a necessidade da sua inclusão ao decriptar.

Tendo todas as informações necessárias, as últimas coisas que precisam ser feitas são colocar a hash em um arquivo .txt pois essa também não pode ser inserida na linha de comando, e repetir o uso do comando `grep -E '^.{6}$' /usr/share/wordlists/rockyou.txt > list.txtpra` um pouco diferente, pra reduzir um pouco o escopo da wordlist e economizar tempo.

Feito isso, usando `hashcat -a 0 -m 1800 hash.txt list.txt` obtemos a resposta **“waka99”.**

Por último, temos uma resolução bem simples. Utilizando o hash-identifier descobrimos que se trata de um SHA-1. No entanto, sabemos que essa hash utiliza o salt fornecido no desafio (tryhackme), então analisando o mesmo site anterior, vemos que existe uma HMAC-SHA1 (key = $salt) com código `160`.

Dessa forma, utilizando `hashcat -a 0 -m 160 e5d8870e5bdd26602cab8dbe07a942c8669e56d6:tryhackme /usr/share/wordlists/rockyou.txt` obtemos a última resposta do CTF, **“481616481616”.**

E enfim acabamos... Valeu pela paciência aí, espero ter ajudado hehe 🥳👍