BRUTE FORCE SENHAS LINUX

Neste artigo irei falar sobre brute force de senhas no Linux, quando se fala de quebrar senhas já imaginamos coisas de filmes onde o hacker aperta um botão e o software descobre caractere a caractere até achar a senha, mais na verdade não é bem assim que funciona.

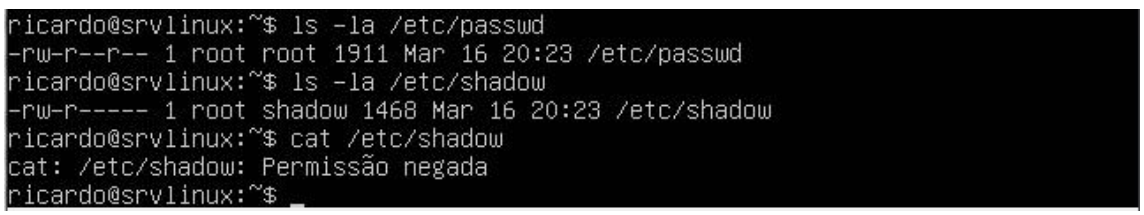
Hoje o ataque de brute force mais bem sucedido são baseados em wordlist, nessa wordlist irá conter palavras do dicionário, sequencias numérica, sequencias do teclado, sequencias do alfabeto, nomes comuns, times, países, palavrões etc.

Um bom hacker consegue deduzir e criar uma boa wordlist, também existe os dumps que vazam na internet de grandes sites no qual possuem varias senhas de muitos usuários.

Tudo que um programa de brute force faz é testar as senhas da wordlist até que alguma seja valida, concedendo o acesso.

Antigamente o Linux armazenava suas senhas encriptadas no arquivo /etc/passwd porem o arquivo /etc/passwd é acessível por todos os usuários do sistema, o que era uma falha pois um usuário podia ver a senha encriptada dos outros usuários e poderia fazer um brute force.

Então criaram o arquivo /etc/shadow no Linux no qual é acessível somente pelo root e mantem a senha criptografada usando um salt.



Salt cria uma sequencia aleatória na hora de criar a senha e faz a criptografia de acordo com aquele salt gerado, isso faz com que 2 senhas iguais tenham um hash diferente.

```
teste:$6$1StbwTiV$JgZkUyFn7SqjJuxKh3L7S4knfQJPZoJH45U/.IzeiMWSQ9r5VxcuP4gNu3KKhq
Bmg8nby7Y7gItZVzMg5Xb.40:16145:0:99999:7:::
teste1:$6$DSuClmrS$8XabOOegY.HyEHT.xDPhNXnuUMIt1WwFzVNq3T3SU4yNz7HUKxHlsMhsv1fsl
EkMDs9McrvFwhij9fS8qtP9u.:16145:0:99999:7:::
root@srvlinux:/home/ricardo#
```

Antes vamos entender como o shadow armazena as senhas.

```
teste:$6$1StbwTiV$JgZkUyFn7SqjJuxKh3L7S4knfQJPZoJH45U/.IzeiMWSQ9r5VxcuP4gNu3KKhq
Bmg8nby7Y7gItZVzMg5Xb.40:16145:0:99999:7:::
```

Cada campo é separado por " **:** " e o primeiro campo é o usuário, o segundo campo começa no **$** e vai até o 40 neste exemplo esse é o hash criptografado, os outros campos refere-se a data de expiração da senha etc

Ainda no campo de senha ele possui outro separador o " **$** ", como podemos notar de **$6$1StbwTiV$** é o salt sendo o **$6$** o id do salt e **1StbwTiV$** o salt, o resto é a senha.

O campo ID indica qual tipo de criptografia, você pode ver isso dando um man crypt.

```
ID   | Method
-----------------------------------------------------------
1    | MD5
2a   | Blowfish (not in mainline glibc; added in some
     | Linux distributions)
5    | SHA-256 (since glibc 2.7)
6    | SHA-512 (since glibc 2.7)
```

No nosso caso estamos trabalhando com tipo 6 = SHA-512.

Se criarmos 2 usuários um chamado teste e o outro teste1 ambos com a senha 102030 eles vão ter hashs diferente, pois o salt era diferente na hora de criar a senha.

```
teste:$6$1StbwTiV$JgZkUyFn7SqjJuxKh3L7S4knfQJPZoJH45U/.IzeiMWSQ9r5VxcuP4gNu3KKhq
Bmg8nby7Y7gItZVzMg5Xb.40:16145:0:99999:7:::
teste1:$6$DSuClmrS$8XabOOegY.HyEHT.xDPhNXnuUMIt1WwFzVNq3T3SU4yNz7HUKxHlsMhsv1fsl
EkMDs9McrvFwhij9fS8qtP9u.:16145:0:99999:7:::
root@srvlinux:/home/ricardo#
```

Agora você deve estar se perguntando e como irei fazer bruteforce nesse caso? Eu criei uma ferramenta onde você coloca o hash e o salt a ser quebrado e ela testa cada senha aplicando a criptografia de acordo com o salt e comparando os hashs, se a senha estiver contida na wordlist ela será quebrada.

Apresento-vos o Loncrack escrito em c:

Para compilar use:

```
root@srvlinux:/home/ricardo/loncrack# gcc loncrack.c -lcrypt -o loncrack
```

Para rodar use:

```
loncrack  loncrack.c  wl.txt
root@srvlinux:/home/ricardo/loncrack# ./loncrack wl.txt
Digite o Hash completo
```

Digite o Hash completo e o salt:

```
root@srvlinux:/home/ricardo/loncrack# ./loncrack wl.txt
Digite o Hash completo
$6$Fq2QZBJm$8SLC5Feono1Z8DeRVJmd1srJuWNRf3/yWISLSdXG5yQjao.ibWsq0vIYRen09pr5ooS2
T6JyaIg/9ycEB6NIP0
Digite o Salt
$6$Fq2QZBJm$
```

Após isso o programa começa o brute force:

```
Testando.. Auxiliares
Testando.. avalon
Testando.. avaluava
Testando.. avancini
Testando.. avareias
Testando.. avatar
Senha encontrada: 4v4t4r
root@srvlinux:/home/ricardo/loncrack#
```

Você pode baixa-lo no seguinte endereço:

http://www.desec.com.br/tools/loncrack.zip

Abaixo segue o código fonte escrito em C:

```c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[])
{
        FILE *arq;
        arq = fopen(argv[1],"r");

        char senha[75];
        char salt[25];
        char comp[100];
        char *result;

        printf("Digite o Hash completo\n");
        scanf("%s", comp);

        printf("Digite o Salt\n");
        scanf("%s",salt);

        int f = 0;

        while(fscanf(arq, "%s", &senha) != EOF)
        {

result = (char *) crypt(senha, salt);
        if (strcmp(comp,result)==0){
        printf("Senha encontrada: %s \n", senha);
        int f = 1;
        return(0);
        }else{
        printf("Testando.. %s \n", senha);
        }
}if(f==0){
printf("Senha não encontrada..\n");
}
```

Como podem ver é um código bem simples, ele abre a wordlist, faz uma leitura linha a linha, pega a palavra + o salt passado e usa a função crypt para gerar um hash depois compara esse hash com o que você quer descobrir, se forem iguais ele para e mostra a senha.

Espero que gostem e quem quiser contribuir para melhorar a ferramenta pode entrar em contato comigo em ricardolongatto@gmail.com.

Ricardo Longatto