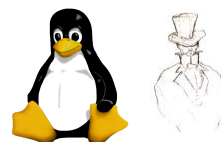


# Capturando Senhas Fracas do Sistema



"Utilize a força, olhe os códigos fontes"

1. *Introdução*
2. *John the Ripper*
3. *Obtendo o pacote*
4. *Instalação e Configuração*
5. *Realizando quebra senhas*
6. *Conclusão*
7. *Referências*

## 1. Introdução

Na administração de um sistema a senha é o ponto mais fraco do sistema de segurança, pois é criado na sua maioria das vezes por quem não possuem formação em segurança de computadores.

Devemos ter como aliados na segurança de sistemas regras básicas na criação de senhas, tais como:

- Definir um tamanho mínimo de senhas de 8 a 10 caracteres;
- Utilizar letras que sejam maiúsculas e minúsculas;
- Utilizar caracteres especiais como \* / = ! @, etc;
- Utilizar números;
- Fazer com que estas senhas sejam trocadas em intervalos de tempo curtos

Utilizando dessas regras tomamos certas providencias que invalida o uso de programas que realizam força bruta no sistema, assim como o **John the Ripper**. Mesmo que o arquivo de senhas seja capturado e realizada a força bruta no mesmo, de acordo com nossa política de criação de senhas, ela já não poderá ser mais valida pois temos a chance do usuário já ter a trocado.

## 2. John the Ripper

**John the Ripper**, é um software utilizado para testar a fragilidade de senhas dos usuários do sistema. Administradores de sistemas têm que ficar antenado em questões de segurança como essa. Assim nos aliamos a esta ferramenta.

Nascido para sistemas unix-like, hoje já é encontrado para as mais diversas plataformas(Linux, BSD, Windows), e desenvolvido pelo projeto OpenWall, ele é capaz de realizar força bruta em senhas criptografadas em DES, MD4 e MD5 entre outras.

## 3. Obtendo o pacote John the Ripper

A obtenção do pacote é feita através da URL:

<http://www.bindshell.net/tools/johntheripper>

Procurando sempre a versão mais atualizada e estável disponível, para garantirmos a não ocorrência de erros no soft.

#### **4. Instalação e Configuração**

Com o arquivo do pacote em .tar.gz, a instalação é simples. Primeiro descompactamos o pacote com o comando:

```
#tar xvf john-X.X.X.tar.gz , Onde X é a versão do pacote.
```

Criamos na raiz do sistema de arquivos um diretório chamado Ferramentas que conterá os arquivos de configuração e execução.

```
#mkdir Ferramentas  
#cd Ferramentas/john
```

Compilando o John - /src

Neste diretório você irá encontrar o sub-diretório "src", é nele que estão os fontes do software e é lá que você precisa ir para compilar o programa:

```
#cd src/
```

Sua compilação é feita baseada na sua plataforma do processador, para que ele seja mais bem utilizado

Digite make e uma lista com todas as possibilidades será exibida para você.

Escolha a mais adequada e:

```
#make clean linux-x86-mmx
```

Lembre-se de substituir o "linux-x86-mmx" pelo valor que mais se adequar ao seu hardware.

Todos os binários resultantes da compilação serão colocados no diretório run, vá para lá:

```
#cd ../run
```

Para testar o binário que foi gerado e verificar se está tudo correto com ele, digite o comando:

```
#!/john --test
```

Isso irá efetuar um teste de benchmarking com todos os algoritmos de criptografia que o JtR suporta.

Após os resultados do benchmarking, o programa já está acessível a através de sua diretório **run/**.

#### **5. Realizando quebra de senhas:**

Como desafio proposto no NetClass da 4Linux, realizaremos aqui a quebra das senhas de um pedaço do arquivo /etc/passwd de um sistema qualquer. Abaixo temos o trecho do arquivo:

```
===== Arquivo /etc/shadow =====  
root:$1$ZEMDF7mW$6fB63sWqcQvKgl7Uqw92q0:12318:0:99999:7::  
user:$1$9rlswUfm$eqPu0J.o.7Vgp9CW8Dj9L.:12318:0:99999:7::  
=====
```

Tirado em :<http://netclass.4linux.com.br>

Com a obtenção dos dados acima, foi criado um arquivo no formato \*.txt que continha esses dados. O nome dado foi senhas.txt.

```
#pwd  
/Ferramentas  
#touch senhas.txt  
#vi senhas.txt
```

Temos então tudo que seja necessário para realizarmos o desafio proposto. Começamos então a rodar o programa que realizara vários algoritmos para gerar senhas randomicamente.

Iniciando o programa:

```
#cd run/  
#pwd  
/Ferramentas/john/run  
#./john ../senhas.txt  
Loaded 2 password hashes with 2 different salts (FreeBSD MD5 [32/32])
```

A ferramenta agora deixa o cursor uma linha abaixo da resposta ao comando, indicando que esta sendo processado o comando. Podemos ter uma rapida visualizacao do tempo que esta sendo rodado, apertando a tecla enter duas vezes. Assim ele nos retorna o seguinte:

```
guesses: 0 time: 0:00:00:03 3% (2) c/s: 4200 trying: Cracker  
guesses: 0 time: 0:00:00:06 7% (2) c/s: 4089 trying: francesco1
```

após um tempo em execução o mesmo nos retorna uma saída com o usuário comum e o usuário de root com suas respectivas senhas desvendadas.

Para uma visualização após a finalização da ferramenta, bastamos usar o seguinte comando:

```
#pwd  
/Ferramentas/john/run  
#./john -show ../senhas.txt  
root:parabens:12318:0:99999:7::  
user:123:12318:0:99999:7::  
  
2 password hashes cracked, 0 left
```

Concluimos nosso desafio então com as informações abaixo:

**Login: root**

**Senha: parabens**

**Login: user**

**Senha: 123**

## **6. Conclusão:**

Podemos concluir que se possuímos uma política rígida na criação de senhas, podemos diminuir a eficácia de programas do tipo brute force. Assim a idéia de realizar uma auditoria com todas as senhas de usuários do sistema, podemos aumentar a segurança do sistema.

Como assim vemos que possuindo senhas fracas do sistema, podemos em questão de horas quebra-las, como foi o caso do arquivo shadow estudado neste artigo, que em 5 horas foi quebrado com facilidade.

## **7. Referências:**

<http://www.neoteam.com.br/index.php?mod=article&cat=Tuto&article=488>

<http://www.vivaolinux.com.br/>

Autor:

Ranyeles Amorim Martins – LPI ID - LPI000154508

ranyeles\_linux\_at\_yahoo\_com\_br

<http://gnuidea.wordpress.com/>