Exercício: Decifrando Senhas

Adaptado do material do Prof. Dênio Mariz

Procedimentos iniciais:

1. Cada aluno deve logar em uma máquina LINUX e abrir uma janela do Shell.
2. Neste exercício, use o seguinte os arquivos disponibilizados pelo professor: **john.tgz**
3. Faça o download dos programas que usaremos neste exercício e instale-os no seu host com os seguintes comandos:

Comando	Comentário
tar xvzf john.tgz	Descompacta o arquivo **john.tgz**
cd john	Posiciona-se no diretório **john**, onde foram descompactados os arquivos que usaremos neste exercício

Exercício 1: Decifrando senhas do Linux

O "John the Ripper" é uma ferramenta muito usada no Linux por administradores de sistema para inspecionar a "qualidade" das senhas usadas pelos usuários. Mas, essa ferramenta também é usada por usuários para decifrar (roubar) senhas de outros usuários para obter acesso não autorizado. Portanto, é bom conhecer seu mecanismo para poder orientar os usuários na escolha das suas senhas.

1. Vamos verificar a performance do "john", para saber quantas cifras por segundo ele é capaz de testar para cada tipo de algoritmo de criptografia. O resultado dependerá do tipo de computador sendo usado. Ele dará o resultado em cifras por segundo (c/s) para cada tipo de algoritmo (Standard DES, BSDI DES, FreeBSD MD5, OpenBSD Blowfish, Kerberos AFS DES, NT LanManager DES).

   Comando => **./john -test**
   Benchmarking: Standard DES [24/32 4K]... DONE
   Many salts: 9631 c/s real, 9748 c/s virtual
   Only one salt: 9241 c/s real, 9334 c/s virtual
   ...

   Compare os resultados e reflita: qual o algoritmo mais demorado para decifrar? E o mais rápido?

2. Vamos tentar "quebrar" (decifrar) algumas senhas contidas no arquivo **passwd** (no diretório corrente), que é uma amostra de um arquivo de senhas original de uma máquina Linux. Este arquivo contém algumas senhas propositadamente escolhidas para mostrar o tempo necessário para decifrar algumas senhas. Mostre o conteúdo do arquivo **passwd** com o comando abaixo:
   comando=> **cat passwd**
   Observe que o arquivo **passwd** contém as seguintes senhas:

Nome do usuário	Senha	Comentários
alunoXX	alunoXX	Alunos de aluno01 até aluno39 – Senhas fáceis
facil	facil	Senha = nome de login (fácil)
suporte	teste01	Palavra do dicionário com sufixo "01" (fácil)
vendas	raquel	Palavra que pode estar em dicionário (fácil)
financas	money	Palavra do dicionário (fácil)
joao	abigail	Palavra que pode estar em dicionário (fácil)
medio	medio001	Dificuldade média (tam>8)
dificil	xX#@-jacare&27_zz	Senha muito difícil

a. Agora vamos executar o comando para tentar decifrar as senhas. Digite o comando abaixo:
comando => **./john passwd**
b. Observe que as senhas são mostradas à medida que ele consegue decifrar. Ele mostra duas colunas: a primeira é a senha quebrada e a segunda é o login. O processamento continua, pois as senhas mais difíceis demoram mais.

c. Espere uns **10 segundos** e cancele o processamento pressionando CTRL-C. Ele mostra uma mensagem parecida com "***guesses: 39 time: 0:00:00:13 28% (1) c/s: 1897 trying: 4lun0394lun0***". Veja o número mostrado após "***guesses:*** ". Ele indica a quantidade de senhas que ele conseguiu decifrar até agora (no nosso exemplo acima, 39).

d. Observe que ele indicou quantas senhas existem no arquivo **passwd** (veja a mensagem "***Loaded 47 passwords with 47 different salts (FreeBSD MD5 [32/32])***")

e. Observe também que ele detectou automaticamente o algoritmo usado na cifragem das senhas (***FreeBSD MD5 [32/32]***).

3. O processo de "quebra" foi cancelado com CTRL-C, mas ele pode ser retomado do ponto onde parou. Isso é porque o "john" guarda no arquivo **john.pot** a posição do trabalho quando é interrompido. Vamos retomar o trabalho:

Comando => **./john passwd**
Observe que ele não mostra as senhas já quebradas. Ele apenas continua de onde estava. Espere mais uns **10 segundos** e cancele novamente com CTRL-C.

4. Vamos tentar decifrar a senha de um usuário específico (usuário suporte)

Comando => **./john –users:suporte passwd**
Espere até ele terminar. Quanto tempo ele leva para decifrar a senha do usuário suporte? Pelo tempo que ele levou para decifrar, voce acha que esse usuário usou uma senha boa?

5. Vamos ver quais as senhas que conseguimos decifrar até agora:

Commando => **./john –show passwd**

6. Vamos tentar decifrar a senha de um usuário específico (usuário joao)

Comando => **./john –users:joao passwd**
Espere +- 1 minuto. Observe que ele não consegue quebrar nesse tempo que esperamos. Depois cancele com CRTL-C.

7. Veja na tabela do item 1 que existem algumas senhas que ainda não foram quebradas (exemplo: usuários vendas, joao etc). Vamos construir um pequeno dicionário de palavras e usá-lo no processo de quebra. Execute os comandos abaixo (digite exatamente as palavras indicadas seguidas de ENTER e no final pressione CTRL-D):

Commando => cat > meudicionario.txt
laranja
banana
rapadura
raquel
money
abigail
CTRL-D

Veja se o arquivo foi criado olhando a lista de arquivos do diretório:

Commando=> ls –l

Agora veja se o arquivo contém o que você digitou (repita o item 7 caso não tenha dado certo):

Commando => cat meudicionario.txt

8. Agora vamos tentar decifrar as senhas usando o dicionário criado. Digite o comando:
Comando => ./john –wordfile:meudicionario.txt passwd

Demorou para decifrar as senhas dos usuários "joao", "financas" e "vendas"? Como você explica o fato de ele ter conseguido em tão pouco tempo?

Observação: Este exercício mostra que quanto maior e mais refinado for o dicionário usado como base, mais provável é a possibilidade de decifrar senhas comuns. Claro, nós sabíamos quais eram as senhas desses usuários, mas, na prática, usam-se dicionários com milhares de palavras e o procedimento que fizemos foi um exemplo de refinamento do dicionário.

9. Vamos acrescentar nosso dicionário criado ao dicionário padrão do "john the ripper", que fica no arquivo password.lst. Digite o comando:

Comando => cat meudicionario.txt >> password.lst

As próximas tentativas de decifragem não precisarão indicar o dicionário, pois quando você não indica um, ele usa o dicionário padrão (password.lst).

10. Vamos tentar decifrar a senha de um usuário específico (usuário dificil)

Comando => ./john –users:dificil passwd

Espere alguns minutos. Observe que ele não consegue quebrar nesse tempo que esperamos. Para falar a verdade, ele pode levar dias, talvez semanas ou meses para quebrar a senha desse usuário (xX#@-jacare&27_zz).

Moral da Estória

Existe na Internet dicionários já prontos com milhares de palavras de todos os idiomas, gírias, termos específicos por área (medicina, engenharia, advocacia, ...), os quais podem ser obtidos por pessoas que têm interesse nessa "difícil" tarefa de "password cracking". Portanto, cabe a você orientar os usuários a usar senhas eficientes, que misturam letras, números e símbolos, além de terem um tamanho mínimo (digamos, 6 ou 7 caracteres).

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – Campus Campina Grande
Curso: Telemática
Disciplina:Segurança de Redes
Prof. João Henrique

Exercício: Decifrando Senhas – Parte II
Adaptado do material do Prof. Dênio Mariz

O WinZip é uma ferramenta muito usada no Windows para gerar arquivos compactados. Adicionalmente, ele permite que indiquemos uma senha que pode ser usada para cifrar o arquivo compactado, de maneira que a descompactação só possa ser feita mediante a mesma senha.

Procedimentos iniciais:

1. Cada aluno deve logar em uma máquina Windows.
2. Neste exercício, use o seguinte os arquivos disponibilizados pelo professor: fzc.zip
3. Salve o arquivo no diretório C:\SEG\FZC (crie este diretório se não existir).
4. Descompacte o arquivo dentro do diretório.

Exercício: Decifrando senhas do WinZip

O exercício seguinte vai usar a ferramenta "Fast Zip Cracker" (FZC), que é gratuita. Esta ferramenta usa os métodos dicionário e força bruta para tentar decifrar senhas que protegem arquivos gerados pelo WinZip (*.zip).

1. Abra uma janela de comandos MS-DOS (Menu Iniciar->Programas->Prompt do MS-DOS)

2. Mude para o diretório C:\SEG\FZC

3. Crie um arquivo ZIP chamado **readme.zip** dentro do diretório C:\SEG\FZC. Para fazer isto execute os seguintes passos:

a. Abra o Windows Explorer, escolha a pasta C:\SEG\FZC

b. Clique com o botão direito sobre o arquivo README.TXT e escolha a opção "Add to Zip".

c. Clique no botão "Password". Na janela que abre, desmarque a opção "Mask Password", depois posicione o cursor no campo "Password" e digite uma senha de **4 letras minúsculas** (exemplo: **aaxy** ou **casa**). Clique no botão "Ok". Agora clique no botão "Add" para criar o arquivo.

d. Depois que o arquivo **readme1.zip** for criado, feche o WinZip.

4. Vamos tentar decifrar a senha pelo método da força bruta. Na janela MS-DOS digite:
Comando => fzc -mb –nzreadme1.zip -l1-4 -ca

(CUIDADO: a opção "–l1-4" é a letra "L" minúscula seguida de "1-4")

A seguir uma explicação para as opções do comando acima:

–mb => escolhe o método da força bruta

–nzFILEZIP => indica o nome do arquivo ZIP para decifrar (no nosso caso, readme1.zip)

–lx-y => indica que o tamanho mínimo da senha é y e o máximo é x (no exemplo, de 1 a 4)

–cCHARSET => indica os conjuntos de caracteres a usar. CHARSET pode ser:

a = letras minúsculas;

A = letras maiúsculas;

s = espaço;

1 = dígitos (0-9);

! = símbolos (!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~));

= todos os caracteres da tabela ASCII.

Em quanto tempo ele conseguiu decifrar ? (anote o tempo em segundos)

5. Repita o processo do item 3 e crie o arquivo **readme2.zip**, mas desta vez digite uma senha de **5 letras** *mesclando letras maiúsculas e minúsculas* (exemplo: **AbRiL**). Depois tente decifrar com o comando abaixo:

Comando => fzc -mb –nzreadme2.zip -l1-5 –caA

Em quanto tempo ele conseguiu decifrar ? (anote o tempo em segundos)

6. Repita o processo do item 3 e crie o arquivo **readme3.zip**, mas desta vez digite uma senha de **5 letras** *mesclando letras maiúsculas, minúsculas e dígitos* (exemplo: **LoV3s**). Depois tente decifrar com o comando abaixo:

Comando => fzc -mb –nzreadme3.zip -l1-5 –caA1

Em quanto tempo ele conseguiu decifrar ? (anote o tempo em segundos)

7. Repita o processo do item 3 e crie o arquivo **readme4.zip**, mas desta vez digite uma senha de **4 letras** *mesclando letras maiúsculas, minúsculas, dígitos e símbolos* (exemplo: **Ma9@**). Depois tente decifrar com o comando abaixo:

Comando => fzc -mb –nzreadme4.zip -l1-4 –caA!1

Alguns dados para reflexão

A tabela abaixo mostra o tempo máximo que o "Fast Zip Cracker" leva para decifrar uma senha do WinZip (benchmark feito em um computador Pentium II 600MHz). Observe que o tempo depende do tamanho da senha e do conjunto de caracteres usado. A tabela é fornecida junto com o software (arquivo BFORCE.TXT).

Código do conjunto	Descrição do conjunto de caracteres	Tamanho do conjunto	Tamanho da senha			
			4	**5**	**6**	**7**
1	0-9	10	<1 s	<1 s	<1 s	5 s
A	A-Z	26	<1 s	5.9 s	2.6 min	1.1 horas
a	a-z	26	<1 s	5.9 s	2.6 min	1.1 horas
!s	Espaço + símbolos !"#$%&' ()*+,./:;<=>?@[\]^_`{|}~)	33	<1 s	20 s	10.8 min	5.9 horas
aA	A-Z, a-z	52	4 s	3.2 min	2.7 horas	5.9 dias
aA1	A-Z,a-z,0-9	62	7 s	7.6 min	7.9 horas	20.4 dias
aA1s!	A-Z,a-z,0-9,espaço,símbolos	95	41 s	1.1 horas	4.3 dias	1.1 anos
#	Toda a tabela ASCII	254	35 min	6.1 dias	4.3 anos	1081 anos

Olhando a tabela acima responda:

a) Qual o tempo máximo de decifragem de uma senha de tamanho 4, seja qual for o conjunto se caracteres usado?

b) Compare o tempo de decifragem de uma senha que usa apenas letras (código **aA**) e as que usam letras e números (código **aA1**)

c) Compare o tempo de decifragem de senhas de tamanho 6 e 7 que usam letras, números e símbolos (código **aA1s!**)