

Força Bruta

Auditoria de Senhas
Crackeando Senhas

O conceito de Intrusão

- ❑ **Ameaça ou Tentativa** (quando o invasor pula o muro).
 - ❑ **Ataque** (concretiza o arrombamento).
 - ❑ **Invasão** (quando obtém sucesso).
-

Para concretizar um Ataque

- **Obter meio de acesso não autorizado** a um sistema remoto com configurações padrões.
-

Força Bruta

❑ ***Auditando*** ou ***Crackeando*** Senhas.

❑ Força Bruta para *crackear* senhas em **Serviços:**

- POP, FTP, Telnet, Proxy-Web,
 - Web Servers, roteadores ou SO
-



Força Bruta para Auditar ou Crackear Senhas

Auditando Senhas

- ❑ Muitas vezes, as senhas são consideradas o lado mais fraco em uma política de segurança.
 - ❑ É da natureza humana procurar a solução mais fácil para um problema.
 - ❑ Usuários tendem a não criar senhas longas e complexas. Pois é difícil de lembrar.
-

Auditando Senhas

- ❑ Muitas vezes tendem a criar senhas com algo no seu ambiente.
 - ❑ Isso torna fácil para um invasor deduzir uma senha, ou fácil para um decifrador de senhas determinar essas senhas fáceis de lembrar.
-

Auditando Senhas

- ❑ A maioria das empresas ainda conta com senhas, como o único mecanismo de autenticação para acesso aos recursos de seus sistemas.
 - ❑ Responsabilidade da equipe de segurança: **garantir que as senhas correspondam a um requisito mínimo de segurança.**
-

Auditando Senhas

❑ **Contrameditada:** o uso de verificadores de senha ou ferramentas de auditoria de senhas para reforçar políticas de senha.

- ajuda a reduzir o risco imposto por senhas mal escolhidas.

- Exemplos:

- Jack Cracker (mais clássica)
 - Nutcracker (Escrito em Perl)
 - **John the Ripper**
-

Usando John the Ripper

- ❑ Alternativa ao Jack Cracker.
 - ❑ Bem mais rápido e sofisticado que o Jack Cracker.
 - ❑ Favorito de muitos *script kiddies* e *crackers*.
 - ❑ É o preferido para auditoria de senha.
 - ❑ Escrito em linguagem C.
-

John the Ripper

- ❑ A maioria dos sistemas usa MD5, ao invés de DES.
 - ❑ Pode ser configurado para usar o tempo inativo do processador, para decifrar sessões.
 - ❑ Permite a restauração de sessões de decifração.
-

John the Ripper

- ❑ Possui uma variedade de conjuntos de regras configuráveis.
 - ❑ Qualquer software de vulnerabilidade de segurança deve ser instalado numa máquina que não seja de produção, pois este software possibilita a qualquer usuário, a decifragem de senhas do sistema.
-

John the Ripper

- ❑ Caso precise, usar permissões bem restritas, para os **arquivos de saída** e para o **arquivo usado para fazer auditoria**, como permissões **700**, com permissão de propriedade de *root*.

 - ❑ Download em:
<http://www.openwall.com/john>
-

John the Ripper – version 1.x

- ❑ Copie para o seu diretório de trabalho e descompacte:

```
> ./tar xzf john-1.x.tar
```

- ❑ Será criado um diretório: `john-1.x`

- ❑ Dentro do qual deverá ser acessado o subdiretório: `/src`
-

John the Ripper – version 1.x

- ❑ `>cd john-1.6/src/`
 - ❑ Será exibida a lista de possíveis compilações, mostrando ser uma ferramenta multiplataforma:
`> ./make`
 - ❑ `/usr` diretório Linux dos arquivos de perfis e configurações pessoais dos usuários.
-

John the Ripper – version 1.x

- ❑

```
>./tar xzvf john-1.x.tar.gz  
-c /usr/local/src
```
 - ❑ Entre no diretório onde foi descompactado e leia o arquivo `readme`:

```
>cd /usr/local/src/readme  
>less readme
```
 - ❑ Dentro do `readme` tem as instruções de como proceder a instalação.
-

John the Ripper – version 1.x

- ❑ Compilando o John ...
 >cd ./scr

 - ❑ >make ... (para a compilação, deve ser indicada a plataforma na qual será instalada John).

 - ❑ Em Linux:
 >make linux-x86-any-elf

 - ❑ >make install
-

John the Ripper – version 1.x

- ❑ Depois da instalação, os arquivos executáveis estarão no diretório:
`/usr/local/john-1.x/run/`
 - ❑ Depois de instalado, pode-se testar as senhas:
`>cd ../run`
 - ❑ `/etc` diretório que contém os arquivos de configuração de utilitários ou programas do sistema
-

John the Ripper

- ❑ shadow – arquivo de senhas do sistema Linux.
 - ❑ Testar as senhas na forma básica de uso do John: `> ./john /etc/shadow`
-

John the Ripper

- ❑ Exemplificando o modo `single` :

```
> ./john -single /etc/shadow
```

- ❑ Quebrando a senha de um usuário específico: `livianvital`

```
> ./john -show -users:livianvital  
/etc/shadow
```

John the Ripper

□ Extrair a senha do shadow:

```
> ./unshadow /etc/passwd  
/etc/shadow > <arquivo-de-senha>
```

Argumentos do John

- ❑ `> ./john -single /etc/shadow`
utiliza as informações de login como base para a wordlist. Forma de simples de cracking.

 - ❑ `> ./john -wordfile:/temp/dictionary.txt
/etc/shadow`
modo de wordlist, serão usadas apenas as palavras contidas no arquivo `dictionary.txt`
-

Argumentos ...

- ❑ `> ./john -rules /etc/shadow`
Habilita regras para o modo de wordlist.
 - ❑ `> ./john -incremental Modo`
poderoso de cracker baseado em
combinações.
 - ❑ `> ./john -external Modo de`
combinação que possibilita a utilização de
definições externas.
-

John the Ripper

- ❑ Em **uma situação ideal**, não convém decifrar o arquivo shadow (arquivo que contém as senhas criptografadas) de uma máquina, na mesma máquina em que se encontra o arquivo shadow.
-

John the Ripper

- ❑ Se precisar executar o John the Ripper na mesma máquina, cuidar com o arquivo `john.pot` no diretório `install directory/john-1.x/run/` **john.pot**
 - ❑ É em `john.pot` que estão todas as senhas decifradas.
 - ❑ Usar este arquivo com permissões restritivas ...
-

John the Ripper

- ❑ Dicionário de palavras (supostas senhas) com 2.290 palavras
... /john-1.x/run/**password.lst**
 - ❑ Para ampliar o dicionário, fazer *download* de outros dicionários, e concatenar ao dicionário *default*. Usar esse último como padrão.
-

John the Ripper

- ❑ Se quiser usar uma lista de palavras diferente da padrão:

```
> ./john -wordfile:[diretorio/arquivo]
```

- ❑ Interrompendo o processamento do arquivo de senha: `CNTL-C`

- ❑ Para reiniciar a sessão:

```
> ./john -restore [arquivo a restaurar]
```

John the Ripper

- Para mostrar todas as senhas decifradas e usuários associados:

```
> ./john -show /etc/shadow
```

John the Ripper

- ❑ Regra estabelecida para verificar senhas de uma maneira concentrada:

Configurando o arquivo `john.ini` localizado em `install directory/run/`, pode-se configurar conjuntos de regras únicos, dependendo das necessidades.

Documentação sobre regras está em `install directory/docs/RULES`.

John the Ripper

- ❑ Os administradores utilizam Verificadores de Senha (Jack Cracker, Nutcracker, [John the Ripper](#)) **em seu ambiente**, para **auditar as senhas** de seu sistema, **descobrendo senhas fracas** e motivando uma política de senhas fortes.
-

Auditando Senhas com o PAM

❑ **Contramedida:**

Configurar o **SO** para verificar o tamanho e a complexidade de senhas através de módulos de autenticação conectáveis (**PAM – Pluggable Authentication Modules**) fornecidos com a distribuição.

❑ **PAM** é a biblioteca que permite autenticar usuários.

PAM

□ Login Remoto:

- **servidor de autenticação** (a base de usuários não está na mesma máquina do usuário, mas em uma máquina da rede).

PAM

- ❑ Modificar o programa *login* para que ele suporte autenticação remota.
 - ❑ Se surgir um novo algoritmo de criptografia, mais rápido, que gostaríamos de usar, termos que modificar novamente o programa *login*.
-

PAM

- ❑ Num SO, muitos programas (aplicações ou serviços) utilizam algum tipo de autenticação de usuários.
 - ❑ Imagine se esses programas tenham que ser reescritos, cada vez que algum dos critérios de autenticação seja alterado.
-

PAM

- ❑ SUN criou o PAM e liberou através de RFC.
 - ❑ O **Linux** derivou sua implementação do PAM, a partir desse documento.
 - ❑ Configurando o PAM no Linux, o programa (aplicação ou serviço) precisa ser reescrito apenas uma vez, justamente para suportar o próprio PAM.
-

PAM

- A partir daí o programa (aplicação ou serviço) delega a responsabilidade de autenticação para o PAM.
-

PAM

- No caso de se querer mudar o algoritmo de criptografia para senhas, basta que o PAM seja modificado para que todos os programas, passem automaticamente e de modo transparente, a usufruir dessa nova forma de autenticação.
-

PAM

- ❑ É possível configurar a autenticação de forma individual para cada programa (aplicação ou serviço).
 - ❑ Com isso, pode-se ter um usuário usando certos recursos de HW, desde que os mesmos sejam acessados pelo console da máquina. Se o *login* não tiver sido feito pelo console, o acesso ao recurso de HW é negado.
-

PAM

- Nenhum programa (aplicação ou serviço) sabe alguma coisa sobre recursos de HW. Eles não precisam saber!
 - O PAM se encarrega disso.
 - O PAM vai além da autenticação.
-

PAM

- Os módulos do PAM podem ser de quatro **tipos**:
 - auth
 - account
 - passwd
 - session
-

PAM – Tipo auth

- ❑ Tipo de módulo que **verifica se o usuário é mesmo quem ele diz ser.**
 - ❑ Pode pedir apenas o ***username*** e uma ***password***.
 - ❑ Ou usar **biometria**: autenticar através da impressão digital, imagem da retina ou impressão de voz.
-

PAM – Tipo account

- ❑ Autorização e Acesso
 - ❑ Verifica se o usuário está autorizado a utilizar o serviço ao qual está se autenticando.
-

PAM – Tipo passwd

- ❑ Usado quando se deseja mudar a senha.
 - ❑ Podem ser adicionados módulos que verifiquem se uma senha é forte ou fraca.
-

PAM – Tipo session

- ❑ Encarregada de executar o que for necessário para **criar o ambiente do usuário**.

 - ❑ Fornecer **acesso a alguns dispositivos locais**:
 - áudio,
 - CD-ROM,
 - fazer registro de eventos nos arquivos de *log* do sistema SO,
 - ou montar sistemas de arquivos.
-

Exemplo de Módulos PAM

- ❑ **pam-pwdb**

Pode ser usado com todos os quatro tipos.

- ❑ **pam-console**

Normalmente usado como *session*.



Força Bruta em Serviços

Força Bruta em Serviços

- ❑ Técnicas clássicas e “barulhentas”.

 - ❑ A maioria dos sistemas para força bruta em serviços, gera *logs* de tentativas de conexão.

 - ❑ Ferramentas:
 - Sdi.brutus.pl (Melo, S. 2004 p.130)
 - Blaster (Melo, S. 2004 p.130)
 - **Hydra** (Melo, S. 2004 p.131)
-

Força Bruta em Serviços - Hydra

- ❑ Escrita na linguagem C.
 - ❑ Desenvolvida para ambientes POSIX.
 - ❑ Objetivo: descobrir *username* e/ou *password* a partir de um serviço.
 - ❑ Arquivos: `userlist.txt`
`passwd.txt`
-

Hydra

- ❑ Front End `xhydra` em GTK.
 - ❑ Linux, UNIX BSD, Solaris, Mac OS/X e outros UNIX-like.
 - ❑ Windows com **Cygwin** (ferramentas Shell GNU que emulam Linux em Windows), com suporte a IPV4 e IPV6.
 - ❑ Sistemas móveis baseados em processadores ARM (Advanced Risc Machine) (Zaurus, Ipaq).
-

Hydra

- ❑ **Prova o conceito de recursos de segurança**, com a possibilidade de mostrar a facilidade de se **obter acesso não-autorizado** a um sistema remoto, ...
 - ❑ ... dentro de um cenário em que o administrador mantém **configurações padrões de contas e senhas fracas** nos serviços disponíveis.
-

Hydra

- É possível testar os seguintes serviços:

Telnet, FTP, HTTP, HTTPS, HTTP-Proxy, LDAP, SMB, SMBNT, MS-SQL, MySQL, POP3, IMAP, NNTP, ICQ, PCNFS, VNC, SOCKS5, REXEC, SAP/R3, Cisco Auth, Cisco Enable, Cisco AAA.

Hydra

- ❑ SSH e Oracle.
 - ❑ Pode usar a técnica de *Bounce* para força-bruta em aplicação Web utilizando um Proxy-Web mal configurado.
-

Hydra

□ Compilando Hydra:

```
>./configure
```

```
>make
```

```
>make install
```

□ Compilando o Front End GTK

```
>cd hydra-gtk
```

```
>./configure && make && make  
install
```

Hydra

- ❑ Compilando em Palm Pilot e Mobiles baseados em processadores ARM:
 - > ./configure-palm
 - > ./configure-arm

 - ❑ Por padrão, o Hydra será instalado em `/usr/local/bin/...`. Seu binário é “hydra” e o binário do Front End é “xhydra”.
-

Hydra

❑ `hydra <ip-alvo> <def-serviço> <opções>`

❑ Opções Especiais:

opção “-m”

Alguns serviços requerem técnicas de força-bruta com a opção “-m”.

WWW, SSL, HTTP, HTTPS,

Hydra

- Restaurando uma sessão abortada ou travada.
 - CNTL C
 - hydra.restore (arquivo)
 - > ???

 - Performance no uso do Hydra
 - opção “-t”
 - desempenho depende do protocolo.
 - o mais rápido, é geralmente, o POP3.
-

Hydra

- ❑ Outras opções em serviços como:

SMBNT, LDAP, serviços Cisco, SAP/R3

- ❑ O aplicativo **PW-INSPECTOR**:

Utilitário para manipular wordlist, extraíndo de uma wordlist uma segunda wordlist seguindo padrão pré-definido pelos seus parâmetros.

Hydra e PW-INSPECTOR

- Serve para criar outras *wordlists*, quando o atacante sabe o perfil de senha que o alvo utiliza, resultando assim numa redução da lista de senhas (*wordlist*).

Por exemplo: senhas com o mínimo de 6 caracteres.

PW-INSPECTOR

- ❑ Seja uma wordlist com vários tipos de senhas: `words.txt`
 - ❑ Ordenando `words.txt`

```
>cat words.txt | sort | uniq >  
dictionary.txt
```
 - ❑ Extraíndo de `dictionary.txt` apenas as senhas que atendam ao padrão:

```
>cat dictionary.txt | pw-inspector -m  
-c 2 -n > passlist.txt
```
-