

Como prevenir o forkbomb do terminal bash.

O forkbomb é a execução de uma série de instruções que possuem o mesmo mecanismo de ataque de negação de serviço (DDOS): sobrecarregam os recursos do sistema, a memória e o CPU, travando a máquina, através da multiplicação em exponencial das instruções do script.

O ataque ao bloquear a máquina, através da multiplicação de processos no sistema em background, dificulta a chance do usuário dono do equipamento de obter o acesso ao sistema, pois o sistema se torna inoperante e a máquina precisa ser reiniciada para voltar as condições normais.

O forkbomb pode ser uma forma também de paralisar uma rede de computadores, se instalado e executado em várias máquinas, pode ser uma forma rudimentar de ataque.

O forkbomb em shell script é geralmente identificado por esta forma:

```
:() { : | : & }::
```

A forma análoga deste script é:

```
forkbomb() {  
    forkbomb | forkbomb & }; forkbomb
```

Os dois pontos é o nome da função (forkbomb) que é chamada de forma recursiva (corpo da função entre chaves), onde a primeira chamada é passada para a segunda pelo pipe e também é executada em segundo plano (background) no sistema.

Geralmente, quando há um script deste em execução, não há uma forma de parar instantaneamente, a não ser que a máquina já esteja preparada para limitar processos, recursos de CPU e memória, pois assim o usuário, com o

controle do sistema, pode facilmente terminar a execução do código.

Como se preparar para um forkbomb?

A primeira, antes de qualquer execução de scripts maliciosos, de mais rápido acesso, é através do terminal, através da chamada ao comando `ulimit`, comando de acesso ao arquivo `/etc/security/limits.conf`, de forma temporária, realiza o limite de número de processos, memória e outros mais.

Ao digitar `ulimit -u 2500`, o usuário limita seu número de processos a 2500. Para máquinas atuais, limites de número de processo entre 2500 e 5000, são bastante efetivos.

No arquivo descrito anteriormente, a linha deste comando, acrescentando limites ao sistema desde sua inicialização, é colocada desta forma:

```
username soft nproc 2500
```

```
username hard nproc 5000
```

onde `username` é o nome do usuário em questão, podendo ser substituído por um asterisco (\*) para a aplicação para todos os usuários.

Há outras formas de se limitar e definir como os processos serão executados, mas esta é a forma principal.

Limitar CPU e memória pelo `SYSTEMD`, utilizando o Linux `cgroups` (Control Groups), também é possível de ser feito. Uma das formas, pela linha de comando, seria forçar a criação de um arquivo com as limitações impostas e recarregá-las no processo de inicialização do sistema.

Exemplo: `systemctl edit --force user-.slice`

Arquivo:

```
[Slice]  
CPUQuota=75%
```

Exemplo: `systemctl edit --force system.slice`

Arquivo:

```
[Slice]  
MemoryMax=5G
```

Estas formas são melhores que as utilizadas pelo arquivo `/etc/security/limits.conf` por serem mais flexíveis com relação aos valores e unidades utilizadas.

É possível remover o usuário do sistema, na hora do ataque, caso o sistema esteja responsivo ainda, utilizando o comando `ps`:

```
ps -ef | grep nomedo usuário.
```

Uma forma bastante interessante e que evita a execução do script malicioso, impedindo assim o travamento do sistema é através da modificação do arquivo `.bashrc` com a inclusão da seguinte instrução:

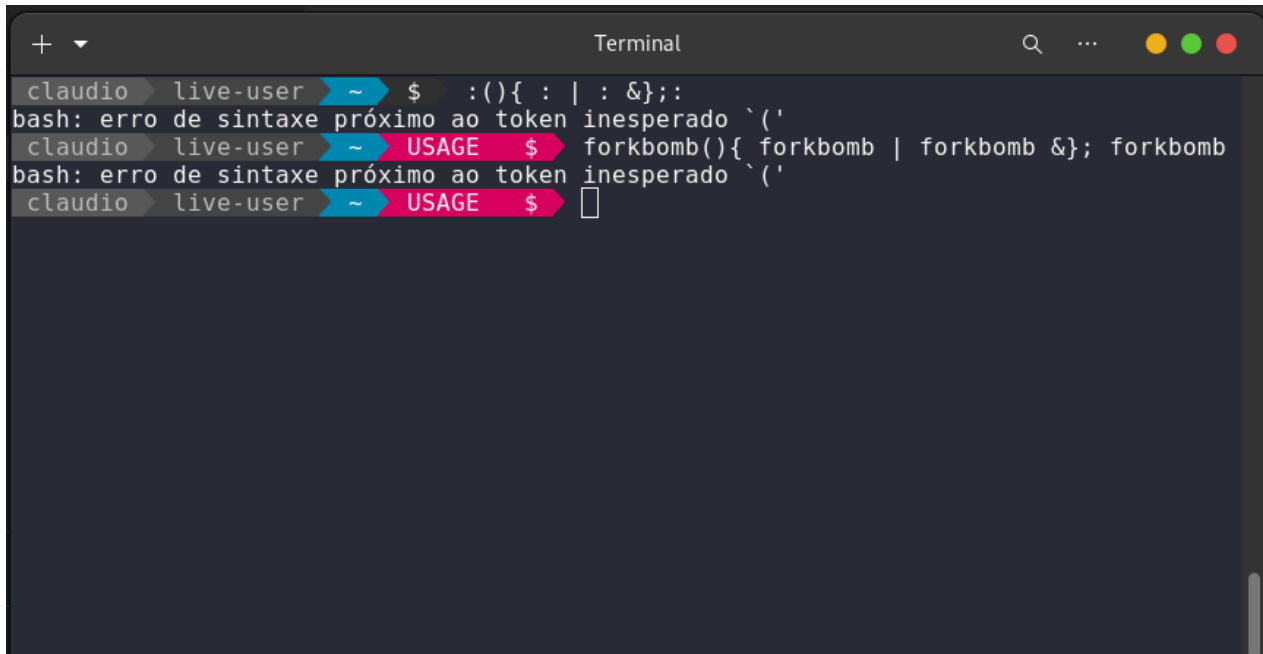
```
alias :="echo 'Forkbombs estão desabilitados!' "  
alias forkbomb="echo 'Forkbombs estão desabilitados' "
```

Há também quem prefira habilitar instruções através do PAM (Módulos de Autenticação Plugáveis), configurando limites de processo por usuário, por grupo ou em todo o sistema. Modificando o arquivo `/etc/pam.d/common-session` e adicionando a linha:

```
session required pam_limits.so  
O sistema estará mais protegido.
```

Estas são as alternativas de proteção para o Forkbomb no Linux.

Espero que estas instruções o ajude a cuidar do seu sistema.

A terminal window titled "Terminal" with standard macOS window controls (yellow, green, red buttons) and search and menu icons. The terminal shows a user named "claudio" at the "live-user" prompt. The first command entered is `$ :(){ : | : & };`, which results in the error message `bash: erro de sintaxe próximo ao token inesperado '('`. The second command entered is `$ forkbomb(){ forkbomb | forkbomb & }; forkbomb`, which also results in the same error message. The third line shows the prompt `$` with a cursor, indicating the command was not executed. The prompt is preceded by a blue arrow and the word "USAGE" in red, suggesting a usage message was displayed.

```
claudio live-user ~ $ :(){ : | : & };  
bash: erro de sintaxe próximo ao token inesperado '('  
claudio live-user ~ USAGE $ forkbomb(){ forkbomb | forkbomb & }; forkbomb  
bash: erro de sintaxe próximo ao token inesperado '('  
claudio live-user ~ USAGE $
```

Até mais,  
Cláudio