**Finger**

**Finger** é um comando de troca de informação sobre utilizadores. O protocolo descrito no RFC é um protocolo muito simples que permite aceder à informação sobre utilizadores de uma maquina remota. (o comando Finger também funciona numa maquina local, efectuando pedidos sobre utilizadores à própria maquina. Mas neste caso não é usado o protocolo de comunicação de redes)

De uma forma genérica o funcionamento do Finger resume-se a efectuar uma conexão à porta 79 do servidor, e a realizar um pedido. O servidor envia a resposta de acordo com os dados do pedido.

Em técnicas hackers, o protocolo finger é usado pra capturar senhas de todos os usários de um sistema para que se possa ter acesso root ao sistema sem a necessidade de quebrar a senha ou usuário por força bruta. Se o pedido for feito em DotIn no sistema, será retornado um erro provocando a quebra do sistema, eis aí o principal motivo de protocolo finger ser quase inexistente na maioria das redes e sua porta protegida diretamente pelo firewall.

Um exemplo de quebra por erro finger aconteceu no ano de 2006 em Salvador, quando um cracker travou o sistema de um supermercado local.

Os dados enviados são em formato ASCII com os códigos de CR e LF no final.

**O RFC**

O RFC usado para implementar este trabalho foi o [RFC 1288](http://tools.ietf.org/html/rfc1288), escrito em Dezembro de 1991. Este RFC actualiza os RFCs 1196, 1194 e 742.

**O Buffer formatado**

O Buffer que recebe os resultados do finger, pode ser imediatamente impresso na tela sem necessidade de tratamento, uma vez que os dados já se encontram formatados.

**Implementacao do cliente Finger**

// - 0 finger do ISEP

// para testar - server: gnu.org users: karl, pot

#include <stdio.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <netdb.h>

#include <errno.h>

#include <unistd.h>

#include <string.h>

#include <stdlib.h>

#define BUFLEN 30000

main (int argc, char \*argv[])

{

struct sockaddr\_in target;

struct hostent \*he;

struct servent \*sp;

int sock, nread;

char \*tmp, buffer[BUFLEN];

char server[100], user[20];

int i;

if (argc==1) {

puts ("Sintaxe do Programa: ifinger [user] server\n");

printf ("Servername: ");

fgets (server, BUFLEN, stdin);

tmp = index (server, '\n');

\*tmp = '\0';

printf ("Username: (ENTER=todos) ");

fgets (user, 20, stdin);

}

if (argc==3) {

strcpy (server,argv[2]);

strcpy (user, argv[1]);

strcat (user,"\r\n");

}

if (argc==2) {

strcpy (server,argv[1]);

strcpy (user, "\r\n");

}

if ((he = gethostbyname (server)) == NULL)

{

perror ("ERRO: nao encontra server");

exit (1);

}

if ((sock = socket (AF\_INET, SOCK\_STREAM, 0)) == -1)

{

perror ("ERRO: nao criou socket");

exit (1);

}

if ((sp = getservbyname ("finger", "tcp")) == NULL)

{

perror ("ERRO: nao encontra porta");

exit (1);

}

memset (&target, 0, sizeof (target));

target.sin\_family = AF\_INET;

// target.sin\_port = 79;

target.sin\_port = sp->s\_port;

target.sin\_addr = \*((struct in\_addr \*) he->h\_addr);

if (connect (sock, (struct sockaddr \*) &target, sizeof (target)) < 0)

{

perror ("ERRO: na conexao");

exit (1);

}

nread = strlen (user);

nread = write (sock, user, nread);

nread = read (sock, buffer, BUFLEN);

printf ("Resultado: \n\n %s", buffer);

close (sock);

exit(0);

}

**Os testes**

Para testar o programa que desenvolvemos, tivemos de encontrar um servidor que aceitasse pedidos de finger. Depois de muita pesquisa encontramos que aceita esse tipo de pedidos e foi este o usado para os nossos testes SERVIDOR: gnu.org

Durante o desenvolvimento e os testes encontramos alguns utilizadores do sistema que poderão servir para efectuar testes:

* Utilizadores: jao, pot.
* Existem também muitos utilizadores com o nome “Karl”