Controle de Acesso e de Aniversariantes

Para BrazilFW

Como diversos companheiros solicitaram eu estou disponibilizando minhas "gambiarras". E isso aí: "gambiarra" mesmo!!! Mas já que pediram... não reparem pois é bem arcaico mas eu me divirto bolando minhas próprias soluções para algumas tarefas que imagino.

Espero que sirva de ponto de partida para alguns se aventurarem também a criar... quem sabe num desses nossos delírios não sai uma grande idéia???

Um grande abraço a todos

Por Elton Guedes Rios

Índice

1.1 Introdução

Conheci o Coyote no início de 2004, através de um tutorial do Carlos Morimoto (www.guiadohardware.net) e confesso: foi "paixão a primeira vista". Tudo devido a simplicidade com que consegui gerar o disquete e colocar pra funcionar meu primeiro servidor. Méritos para os que desenvolviam o projeto mas também, e sem dúvidas, ao Morimoto que, de uma maneira muito simples e clara, conseguiu conduzir através de seu texto, um usuário leigo como eu, ao êxito numa experiência desconhecida. Um usuário mais avançado teria apenas passado os olhos no texto e pronto... já estaria resolvido. Eu não... imprimi, li e reli. Ele orientava tudo, desde a montagem do hardware à configuração final. Fui seguindo criteriosamente (ainda recorri a outras pesquisas prá tirar uma ou outra dúvida que ainda surgia) e no fim deu tudo certo.

É assim que gosto de fazer minhas experiências... tendo alguns exemplos, preferencialmente os que mostrem o passo-a-passo, pois como não tenho conhecimentos profundos ou sólidos em linguagens, hardware, etc , preciso ir naquela da "tentativa-e-erro". Mas é legal, me divirto com isso. E é assim que faço minhas "gambiarras" e é assim que procuro escrever minhas "receitas": simplicidade, criatividade, improvisação. Deixo as grandes descobertas e experimentos com os *donos do conhecimento*... fico apenas com o "feijão-com-arroz".
Minha pretensão ao escrever este e outros tutoriais nunca foi de **ensinar** alguém a fazer mas sim ENCORAJAR, mostrar que com pouco conhecimento e uma boa dose de vontade e curiosidade a gente consegue muitas coisas. Portanto, eles são escritos para os iniciantes. Testo estes "experimentos" a cada dia e é possível que além de aprimoramento eles ainda careçam de acertos... fique esperto. ;-)

Se você já possui bons conhecimentos deve achar tudo adiante muito chato e uma "gambiarrice danada" mas se ainda assim quiser continuar, esteja a vontade. As soluções relatadas aqui nem sempre serão as mais interessantes e muito provavelmente alternativas mais técnicas, clássicas e "limpas" poderão ser implementadas. Em função do meu pouco conhecimento tive que "lutar com minhas armas", mas sempre que aprendia algo novo eu revia tudo o que já estava pronto e melhorava um pouco mais. Já tive por exemplo um script para liberar e outro prá bloquear, e isso para cada usuário. Prá quem conhece é simples, mas eu tive que passar por esta etapa. Cheguei neste caso, a reduzir 14 scripts a um só. Ainda assim tem muita coisa a ser otimizada e melhorada neste meu trabalho. Ajude-nos corrigindo eventuais erros, aprimorando nossos conhecimentos, e lembre-se: meu papel é encorajar os iniciantes, o seu (se conhece mais) é ENSINÁ-LOS.

1.2 Meu BFW

BFW: 2.29beta3 | **CONEXÃO:** ADSL Velox 2Mb
HARDWARE: Pentium MMX233MHz | 32Mb | HD20Gb | 2xRTL8139C | DWL-G520
LAN: 7pcs | **SWITCHER:** Encore ENH908NWY 10/100mbps | **MODEM:** SpeedStream 5200
ADD-ON: IPacc | IPUpdate | HDParm | SCP | IMSNiff | Altivore[1] | Squid | NTM | BMP | MyMail | ArvWireless[2]

1 - Altivore: ainda não rodou 100%
2 - Wireless: ainda não foi implantada

1.3 Minha Rede

Minha rede é doméstica, formada hoje por um notebook (minha máquina) e mais cinco PCs (três filhos e dois sobrinhos). Toda a rede é cabeada sendo que um dos PCs dos sobrinhos fica na casa ao lado. Um segundo está aqui em casa mesmo, em caráter provisório: assim que conseguir instalar a parte wireless ele deve ficar a cerca de 1 km, mas com visada direta. Tem ainda mais um (fica do outro lado da rua), que também está aguardando a parte wireless para poder então entrar na nossa rede, e também é de um sobrinho.
Como podem ver, é uma rede doméstica mesmo e totalmente familiar. Os usuários vão dos 10 aos 18 anos (ei... eu tô fora desta faixa... rs). Como tudo na vida, aqui também tem que ter controle senão vira "bagunça". Tem a questão de custo (energia elétrica), e o pior: impacto na questão dos estudos (se deixar eles ficam 24 horas plugados).
Poderíamos apenas "baixar decretos" mas sabem como é... são todos brasileiros... nenhum deles se chama Gérson, mas também "gostam de levar vantagem em tudo, certo"? Chegamos a pensar em criar uma sala

somente para os micros mas ameaçaram uma rebelião (rs). Os micros permaneceram nos quartos e aí só nos restou mesmo usar os recursos da informática prá tentar "dosar" o uso.
Teve um momento em que distinguíamos a navegação pura e simples do uso de ORKUT e MSN. Duas pragas virtuais que aprisionam nosso filhos e as vezes até nós mesmos (ufff... nem sei como usa isso... desde que me libertei do ICQ que dispensei este tipo de utilização... hoje até penso em testar este tal de MSN apenas para facilitar pequenos, breves e específicos contatos). Assim, determinados horários eles podiam navegar mas não acessavam estes dois serviços. Isso era feito através de uma combinação de Squid (ACLs), regras iptables e um pouquinho de nada de comandos em shell scripts. Isso rendeu o tutorial sobre ACLs, que está na seção Knowledge Base. Hoje, resolvemos fazer simplesmente "navega"ou "não navega" e por isso estamos sem regras acls específicas e concentramos tudo no firewall simplificado. Isso limita um pouco mas foi minha opção prá poder aprender um pouco de shell script "na marra". Um dos problemas (talvez tenha solução mas ainda não descobri) é que quando bloqueio um micro fico sem acesso ao servidor através dele, seja prá rodar o putty ou webadmin. Deve ter uma forma de bloquear tudo deixando apenas o acesso a porta 8180... confesso que "perdi esta aula" :-) Por esta razão tenho que manter um micro sempre liberado (no caso, o meu).

1.4 Criando os experimentos

Inicialmente eu fui trabalhando em cima do MENU (/usr/sbin/menu). Foi nele que comecei a brincar de shell script. Andei travando o Coyote algumas vezes e já na era do BFW isso aconteceu bem menos. Eu diria que assimilei alguns conceitos que minimizaram este problema, mas tudo fazia parte do aprendizado.

Primeiro eu reformulei todo o menu, traduzindo para o português e criando algumas opções adicionais que facilitavam a minha vida. Eu pesquisava e lia muita coisa sobre shell script, mas nunca estudava profundamente... eu apenas ia aprendendo o básico e adaptando. Pegava um trecho de um script, juntava com outro, e nessa brincadeira ia descobrindo coisas e aprendendo mais.

1.5 Script para controle de acesso

Eu fiquei pensando em como poderia efetuar bloqueio e/ou liberação dos usuários, individualmente ou em grupo, conforme as "normas da casa". Isso teria que ser feito automaticamente mas também deixar a opção de intervenção manual (vá que um dos filhos tenha um trabalho escolar e necessite realmente acessar fora dos horários previamente estabelecidos).
Nossa ação de bloqueio e liberação ocorre sempre no arquivo /etc/coyote/firewall. Como isso é feito?

Primeiramente observemos que o arquivo mencionado vem, como default, com as seguintes linhas:

```
# Firewall Access Configuration File
#
# This file contains entries in the following format:
# type active permit|deny protocol source[/mask] destination[/mask] port
#
# type = access # Control access THROUGH the Firewall
# type = admin # Control access TO the Firewall
# active = Y or N

access N deny all 192.168.0.44 any all #Examplo - Deny internet access to this IP
access N deny all 192.168.0.50/23 any all #Example - Deny internet access to these IP
access N deny tcp any any 21 #Example - Deny access to FTP sites
```

Usando a opção Firewall Simplificado no WebAdmin, seleciono a opção **Nega o acesso a todos os usuários internos** e faço o cruzamento dos MACs e IPs (cadastro cada IP e seu respectivo MAC logo na seqüência (*figura 1*).

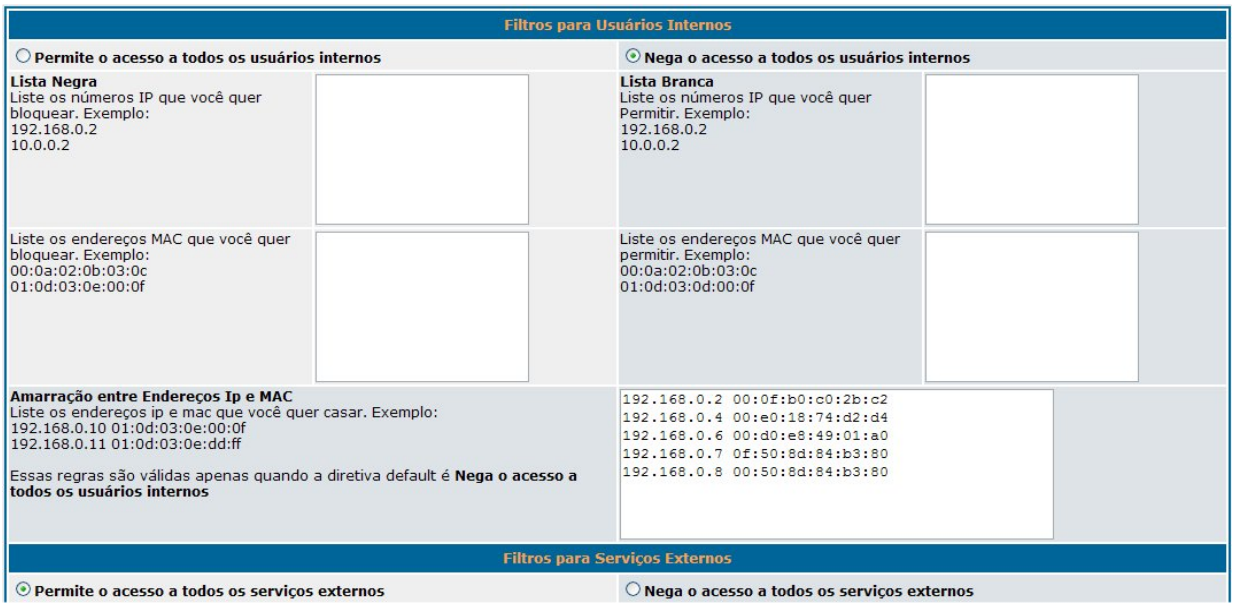


Figura 1

Quando se faz isso serão geradas as linhas abaixo(no /etc/coyote/firewall), que significam que, todos os IP/MAC que estiverem aqui relacionados (e que respeitem as combinações cadastradas) estarão com acesso liberado.

```
match_ip_mac 192.168.0.2 00:0f:b0:c0:2b:c2
match_ip_mac 192.168.0.12 00:11:D8:4D:22:F7
match_ip_mac 192.168.0.4 00:e0:18:74:d2:d4
match_ip_mac 192.168.0.6 00:d0:e8:49:01:a0
match_ip_mac 192.168.0.7 0f:50:8d:84:b3:80
match_ip_mac 192.168.0.8 00:50:8d:84:b3:80
```

O que eu fiz então foi acrescentar algumas informações (grupo e nome), mas atenção: da forma como demonstro aqui vale prá versão 2.29beta3. Na 2.28 era um pouco diferente pois as linhas eram “allow_mac” (liberado) ou “block_mac” (bloqueado) ao invés de “match_ip_mac” (se existe a linha está liberado, senão está boqueado, desde que marcada a opção “*Nega o acesso a todos os usuários internos*”**)**

```
# CONTROLE DE ACESSO
match_ip_mac 192.168.0.2 00:0f:b0:c0:2b:c2 # xxxxx # Gato
match_ip_mac 192.168.0.12 00:11:D8:4D:22:F7 # Todos # Matheus
match_ip_mac 192.168.0.4 00:e0:18:74:d2:d4 # Todos # Wellington
match_ip_mac 192.168.0.5 00:0c:6e:e2:6e:45 # Todos # Anna_Paula
match_ip_mac 192.168.0.6 00:d0:e8:49:01:a0 # xxxxx # Doriel
match_ip_mac 192.168.0.7 0f:50:8d:84:b3:80 # Todos # Willian
match_ip_mac 192.168.0.8 00:50:8d:84:b3:80 # Todos # Jeifferson
```

Se eu fizer alterações diretamente nas caixas do Firewall Simplificado ele vai refazer o arquivo e vai eliminar as informações adicionais que coloquei, por isso não faço alterações ali (creio ser possível modificar o script de reescrita do arquivo de forma a preservar o que não queremos que seja apagado, mas isso é um “dever de casa”).

Mas onde faço as alterações? Através de um script que chamei de **egr.acesso**:

A sintaxe para este script inclui três argumentos: o primeiro é o nome do usuário, o segundo é a ação (liberação ou bloqueio) e o terceiro é só o ponto de onde está sendo chamado, só prá informação de log.

```
 #!/bin/sh

################################################
# NOME: /usr/sbin/egr.acesso                   #
# FUNÇÃO: Bloqueia internet                    #
# Criado por Elton Guedes Rios, em 03/08/2006 #
#                                              #
# Sintaxe:                                     #
```

```
# egr.acesso $1 $2 $3                          #
#                                              #
# Exemplo:                                     #
# egr.acesso Matheus bloqueio menu             #
#                                              #
# Onde:                                        #
# $1 = Nome do usuário                         #
# $2 = Tipo de ação (bloqueio ou liberação)    #
# $3 = Origem (cron, menu, etc)                #
#                                              #
################################################
```

Já dei muita cabeçada por deixar esta linha abaixo de lado. Hoje eu a incluo em todos os meus scripts. Alguém se habilita a explicar tecnicamente o que ela faz?

```
. /etc/coyote/coyote.conf
```

Faço log prá muita coisa prá que eu possa mais facilmente diagnosticar problemas, então defino o caminho onde serão gravados os logs e algumas variáveis que serão utilizadas.

```
# ------------------------------------------
# DEFINE VARIÁVEIS:
# Caminho para logs
LOG=/var/log/admin.log
# Dia da semana em inglês
DIA=$(date +%A)
# Hora atual
HORA=$(date +%X)
#Data no formato dd/mm/aaaa
HOJE=$(date +%d/%b/%Y)

# ------------------------------------------
# TRADUZ DIA DA SEMANA:
if [ $DIA = "Sunday" ]; then
DIA_SEMANA="Domingo";
elif [ $DIA = "Monday" ]; then
DIA_SEMANA="Segunda";
elif [ $DIA = "Tuesday" ]; then
DIA_SEMANA="Terça";
elif [ $DIA = "Wednesday" ]; then
DIA_SEMANA="Quarta";
elif [ $DIA = "Thursday" ]; then
DIA_SEMANA="Quinta";
elif [ $DIA = "Friday" ]; then
DIA_SEMANA="Sexta";
else [ $DIA = "Saturday" ];
DIA_SEMANA="Sábado"
Fi
```

Agora vamos ver o que este script faz:

```
if [ $2 = "bloqueio" ]; then
        sed -i '/'$1'/s/match/xatch/' /etc/coyote/firewall;

elif [ $2 = "liberação" ]; then
        sed -i '/'$1'/s/xatch/match/' /etc/coyote/firewall;

else
        echo
        echo "Sintaxe: egr.acesso $1 $2 $3 "
        echo "   Onde: $1 (nome_usuário)"
        echo "         $2 (ação: bloqueio ou liberação)"
        echo "         $3 (origem: cron ou menu)"
        echo "Pressione ENTER para retornar ao menu."
        read JUNK

fi
```

Ele verifica qual a ação (segundo argumento) solicitada. Aqui eu descobri o SED, um comando muito bacana e versátil. Se for bloqueio ele vai procurar em todo o arquivo /etc/coyote/firewall pelas linhas que contenham o argumento $1 (que é o nome do usuário) e nelas irá substituir a palavra “match” (cada uma que for encontrada) por “match”. Percebem a “gambiarra”??? É que assim o BFW não vai aplicar a regra a este IP/MAC, ou seja, nesta condição (linha iniciada com a palavra “xatch”) este usuário estará bloqueado (óbvio,

pois dexa de estar relacionado dentre os liberados). O inverso para o caso da ação solicitada ser a liberação: ele troca “xatch” por “match”.

```
/etc/rc.d/rc.firewall
echo $HOJE-$DIA_SEMANA-$HORA - Forçado $2 para $1 pelo $3 >> $LOG
```

Feito isso ele recarrega o firewall e registra no arquivo de log... pronto, você já alterou o status do usuário.

1.6 Regras básicas

Desenvolvido este script eu criei regras básicas no cron atendendo uma programação de liberação/bloqueio ao longo da semana.

```
# CONTROLE DA NAVEGAÇÃO
# Libera todos os dias da semana às 14:00h
0 14 * * 1,2,3,4,5 egr.acesso Todos liberação CRON
# Bloquei todos os dias da semana às 18:00h
0 18 * * 1,2,3,4,5 egr.acesso Todos bloqueio CRON
# Libera todos aos sábados e domingos às 13:00h
0 13 * * 0,6 egr.acesso Todos liberação CRON
# Bloqueia todos aos sábados e domingos às 15:00h
0 15 * * 0,6 egr.acesso Todos bloqueio CRON
# Libera todos aos sábados e domingos às 18:00h
0 18 * * 0,6 egr.acesso Todos liberação CRON
# Bloqueia todos aos sábados e domingos às 20:00h
0 20 * * 0,6 egr.acesso Todos bloqueio CRON
```

Observem que agora apareceu o usuário “*Todos*”... pois é, é mais uma gambiarra prá poder aplicar a regra a todos os usuários... todos não: meu micro fica de fora disso... hehehe. Quando aplico a palavra “*Todos*” como primeiro argumento (que seria o nome do usuário), o SED vai procurar as linhas que contenham esta palavra e efetuar a substituição de *match*/*xatch*, conforme o caso.

Isto é o que defini como sendo o grupo. Lá nas linhas do /etc/coyote/firewall, quando inseri outras informações eu coloquei um “xxxxxx” para meu IP/MAC, ao invés de “Todos”, assim fico fora destas regras. Eu poderia usar um outro nome... mais tarde ainda vou fazer isso prá ficar mais limpo. Isto é o básico e o que acontece automaticamente. Fora disso, qualquer alteração será feita diretamente no Menu.

1.7 Personalizando o Menu

Como já disse, modifiquei o menu ao meu gosto (*figura 2*) e inclui opções para gerenciar o status dos usuários de maneira que minha esposa pudesse, com facilidade na minha ausência, liberar ou bloquear quem ela desejasse e fosse necessário.

```
+-------------------------------------------------------------------------------+
|*              ...:::( BrazilFW Firewall e Router ):::...                      |
|                                                                               |
| [A] Teste DNS             [B] Status Servidor       [C] Config BFW            |
| [D] Status Interfaces     [E] Status Hardware/SO    [F] Recarrega Firewall    |
| [G] Teste Gateway         [H] Status Hard-Disk      [I] IPs Conectados        |
| [J] Edita CRON            [K] Edita MENU            [L] Altera senha          |
| [M] Gravar                [N] Backup Especial       [O] Gravar e Reiniciar    |
| [P] Desconectar           [Q] Sair do Menu          [R] Reiniciar             |
| [S] Desligar BFW          [T] Salvar e desligar     [U] Edita firewall        |
| [W] Recarrega webadmin                                                        |
+-------------------------------------------------------------------------------+
|               LIBERA                |               BLOQUEIA                  |
+-------------------------------------------------------------------------------+
|             (AA) Matheus            |          (GG) Matheus                   |
|             (BB) Wellington         |          (HH) Wellington                |
|             (CC) Anna Paula         |          (II) Anna Paula                |
|             (DD) Doriel             |          (JJ) Doriel                    |
|             (EE) Willian            |          (KK) Willian                   |
|             (FF) Jeiff              |          (LL) Jeiff                     |
|             (22) TODOS              |          (44) TODOS                     |
|                        (xx) Lista status de acesso                            |
+-------------------------------------------------------------------------------+
```

Figura 2

Mas aí eu resolvi brincar mais com os scripts e inventar... hehehe... isso é divertido. Eu queria definir um tempo para a liberação.

Isso mesmo, digamos que a gente queira liberar um usuário mas vá sair de casa. Ele ficaria liberado até a gente voltar, ou teríamos que colocar, "na unha" como se diz por aqui, uma regrinha no cron. Então eu queria determinar um tempo e que o bloqueio ocorresse ao findar do mesmo, automaticamente. Esse tá uma gambiarra das grandes... hehehe... ainda quero voltar nele um dia e melhorá-lo.

Vamos lá: se eu quiser, por exemplo, liberar o usuário Matheus, seleciono AA. Isso vai executar o seguinte:

```
#####################################
# Opção [AA] - LIBERA MATHEUS:
#####################################
aa | AA) echo
        egr.quer_tempo Matheus
        ;;
```

Como podem ver isso chama um outro script (**egr.quer_tempo**).

```
echo
echo -n "Quer estabelecer um tempo (máx. 120 minutos)? - [S/N] - "
read SIM_NAO
if [ $SIM_NAO = s -o $SIM_NAO = S ]; then
        egr.quanto_tempo $1
elif [ $SIM_NAO = n -o $SIM_NAO = N ]; then
        egr.acesso $1 liberação MENU
else
        exit 0;
fi
```

Ele vai perguntar se quero estabelecer um tempo ou não. Se responder que ***não***, ele carrega o egr_acesso (já comentado) pegando o parâmetro **$1** (nome do usuário), comandando **liberação** e apontando a origem como sendo **MENU**. A liberação durará até que eu mande bloquear. Se eu disser ***sim*** ele chama o **egr_quanto_tempo** (queria fazer apenas um script mas me enrolei e ficou assim... por enquanto). Note que ele também passa o parâmetro $1.

1.8 Temporizando a liberação

Pensei muito em como fazer isso... no ônibus, no helicóptero, no banheiro, no carro, tomando banho, trabalhando... ufffffffaaaaa... não vou conseguir explicar (nem vou tentar) a lógica da coisa... sei que fui raciocinando e testando numa planilha em excel (adoro brincar também com planilhas), e ia testando e comparando... no fim parece que deu certo.

Preparando variáveis:

```
        # DEFINE VARIÁVEIS:
        # Caminho para logs
        LOG=/var/log/admin.log
        # Dia da semana em inglês
        DIA=$(date +%A)
        # Hora atual no formato hh:mm:ss
        HORA=$(date +%X)
        #Data no formato dd/mm/aaaa
        HOJE=$(date +%d/%b/%Y)
        # Pega parte em minuto da hora atual
        PARTE_MINUTO_HORA_ATUAL=$(date +%M)
        # Pega parte da hora da hora atual
        PARTE_HORA_HORA_ATUAL=$(date +%H)
        # Dia do Mês
        DIA_MES=$(date +%d)
        # Número do Mês
        MES=$(date +%m)
```

Traduzindo... só "perfumaria":

```
        # TRADUZ DIA DA SEMANA:
        if [ $DIA = "Sunday" ]; then
```

```
        DIA_SEMANA="Domingo";
        elif [ $DIA = "Monday" ]; then
        DIA_SEMANA="Segunda";
        elif [ $DIA = "Tuesday" ]; then
        DIA_SEMANA="Terça";
        elif [ $DIA = "Wednesday" ]; then
        DIA_SEMANA="Quarta";
        elif [ $DIA = "Thursday" ]; then
        DIA_SEMANA="Quinta";
        elif [ $DIA = "Friday" ]; then
        DIA_SEMANA="Sexta";
        else [ $DIA = "Saturday" ];
        DIA_SEMANA="Sábado"
        fi
        # --------------------------------------------
        # NÚMERO DO DIA DA SEMANA(FORMATO CRON) :
        if [ $DIA = "Sunday" ]; then
        DIA_SEMANA_NR="0";
        elif [ $DIA = "Monday" ]; then
        DIA_SEMANA_NR="1";
        elif [ $DIA = "Tuesday" ]; then
        DIA_SEMANA_NR="2";
        elif [ $DIA = "Wednesday" ]; then
        DIA_SEMANA_NR="3";
        elif [ $DIA = "Thursday" ]; then
        DIA_SEMANA_NR="4";
        elif [ $DIA = "Friday" ]; then
        DIA_SEMANA_NR="5";
        else [ $DIA = "Saturday" ];
        DIA_SEMANA_NR="6"
        fi

        # --------------------------------------------
        # TRADUZ DIA DA SEMANA:
        if [ $DIA = "Sunday" ]; then
        DIA_SEMANA="Domingo";
        elif [ $DIA = "Monday" ]; then
        DIA_SEMANA="Segunda";
        elif [ $DIA = "Tuesday" ]; then
        DIA_SEMANA="Terça";
        elif [ $DIA = "Wednesday" ]; then
        DIA_SEMANA="Quarta";
        elif [ $DIA = "Thursday" ]; then
        DIA_SEMANA="Quinta";
        elif [ $DIA = "Friday" ]; then
        DIA_SEMANA="Sexta";
        else [ $DIA = "Saturday" ];
        DIA_SEMANA="Sábado"
        fi
```

Primeiro ainda será feita mais uma pergunta:

```
clear
echo
echo "Quanto tempo (máx. 120 minutos)?"
read TEMPO
```

Como limitei em no máximo 120 minutos (quero ampliar isso depois) será feita uma verificação. Se o tempo estipulado for maior que isso ele dá uma mensagem e sai do script.

```
if [ $TEMPO -gt 120 ]; then
        echo "O tempo máximo permitido é de 120 minutos."
        sleep 2
        exit 0;
```

Caso contrário ele executa os cálculos:

```
        ## Soma com o tempo previsto a parte em minuto
        CALC_MIN=$(($TEMPO+$PARTE_MINUTO_HORA_ATUAL))
        ## Verifica se o término ainda estará dentro da hora atual
        if [ $CALC_MIN -lt 60 ]; then
                HORA_FINAL=$PARTE_HORA_HORA_ATUAL
                MINUTO_FINAL=$CALC_MIN
        else
```

```
                ## Somar 1 a parte da hora
                HORA_FINAL=$((1+$PARTE_HORA_HORA_ATUAL))
                ## Diminui o resultado do tempo previsto
                MINUTO_FINAL=$(($CALC_MIN-60))
                if [ $MINUTO_FINAL -gt 59 ]; then
                        MIN_FIN=$MINUTO_FINAL
                        MINUTO_FINAL=$(($MIN_FIN-60))
                        HOR_FIN=$HORA_FINAL
                        HORA_FINAL=$(($HOR_FIN+1))
                else
                        echo
                fi
        fi
```

Concluídos s cálculos é efetuada a liberação do usuário normalmente, sem nenhum apontamento do tempo.

```
        ## Libera acesso ao usuário especificado
        echo "Liberando acesso a $1 por $TEMPO minutos..."
        echo
        egr.acesso $1 liberação TEMPORIZADO
```

Na seqüência o registro no log:

```
        # REGISTRA NO LOG:
        echo $HOJE-$DIA_SEMANA-$HORA - Liberando $QUEM por $TEMPO minutos >> $LOG
```

Agora é feita a preparação de algumas variáveis:

```
        ## Acrescenta linha no ROOT (CRON)
        if [ $MINUTO_FINAL -eq 59 ]; then
                MINUTO_FINAL2=$(($MINUTO_FINAL-1))
                # Faz variável prá ser usada com o apagamento da regra no cron
                MINUTO_FINAL3=$((MINUTO_FINAL2+1))
        else
                MINUTO_FINAL2=$MINUTO_FINAL
                # Faz variável prá ser usada com o apagamento da regra no cron
                MINUTO_FINAL3=$((MINUTO_FINAL2+1))
                MINUTO_FINAL4=$((MINUTO_FINAL3+1))
                MINUTO_FINAL5=$((MINUTO_FINAL4+1))
        fi
```

E depois é escrita uma regra no cron que fará o bloqueio após decorrido o tempo determinado:

```
        echo "$MINUTO_FINAL2 $HORA_FINAL $DIA_MES $MES $DIA_SEMANA_NR egr.acesso $1 bloqueio TEMPORIZADOR"
- $HORA_FINAL$DIA_MES$MES-$1 >> /var/spool/cron/crontabs/root
```

Mas seria muito ruim se estas lihas fossem inseridas e depois eu tivesse que vir apagando manualmente (ou o arquivo poderia se tornar grande demais), e por sso pensei em fazê-lo automaticamente. Como? Sei lá... deve existir alguma maneira mais fácil de fazê-lo mas tive que preparar mais uma “gambi”... gero outras linhas no cron que farão a tarefa de limpeza. Estas linhas possuem um identificador ($HORA_FINAL$DIA_MES$MES) que é usado como referência no momento da limpeza, prá garantir que não serão deletadas linhas erradas.

```
        # Acrescenta linha prá limpar regra depois de cumprida
        echo "$MINUTO_FINAL3 $HORA_FINAL $DIA_MES $MES $DIA_SEMANA_NR sed -i '"/$HORA_FINAL $DIA_MES $MES
$DIA_SEMANA_NR egr.acesso $1"/s/$MINUTO_FINAL2 $HORA_FINAL $DIA_MES $MES $DIA_SEMANA_NR egr.acesso $1
bloqueio TEMPORIZADOR/ /' /var/spool/cron/crontabs/root # $HORA_FINAL$DIA_MES$MES-$1" >>
/var/spool/cron/crontabs/root
        # APAGAR ARQUIVOS DE LOG
        echo "$MINUTO_FINAL4 $HORA_FINAL $DIA_MES $MES $DIA_SEMANA_NR grep -v $HORA_FINAL$DIA_MES$MES-$1
/var/spool/cron/crontabs/root > /var/spool/cron/crontabs/root.temp" >> /var/spool/cron/crontabs/root
        echo "$MINUTO_FINAL5 $HORA_FINAL $DIA_MES $MES $DIA_SEMANA_NR mv /var/spool/cron/crontabs/root.temp
/var/spool/cron/crontabs/root # $HORA_FINAL$DIA_MES$MES-$1" >> /var/spool/cron/crontabs/root
```

O cron é recarregado e... pronto. Boa sorte!!! Nos meus testes isso funcionou... espero que continue....hehehe.

```
## Recarrega CRON
        /usr/sbin/cron.reload
```

1.9 Verificando status

Quando quero checar o status dos usuários eu apenas uso o comando grep e ele lista na tela os LIBERADOS e na seqüência os BLOQUEADOS:

```
#####################################
# Opção [XX] - STATUS DE ACESSO:
#####################################
xx | XX) echo
       echo "========================================="
       echo "LIBERADOS:"
       grep 'match' /etc/coyote/firewall | cut -f7 -d" "
       echo "========================================="
       echo "BLOQUEADOS:"
       grep 'xatch' /etc/coyote/firewall | cut -f7 -d" "
       echo "========================================="
       echo
       echo "Pressione ENTER para retornar ao menu."
       read JUNK
       ;;
```

1.10 Controlando no WebAdmin

O menu estava bastante funcional, mas eu queria e aventurar nos scripts cgi... e agora? Só tentando, certo? Então lá fui novamente. Mergulhei em /var/http/htdocs/cgi-bin/

Observei a tabela de Configurações Avançadas de Firewall e percebi que ela apresentava um layout que me atenderia e então resolvi toma-la por base. Fui analisando e primeiramente enxugando, eliminando aquilo que era bem específico a suas funções e deixando basicamente o que era só a montagem da mesma. Fui aos poucos compreendendo seus comandos e assim fui adaptando ao que eu queria. Simples, não?

O arquivo base foi o **firewall.cgi** e foi gerado o **egr.controle.cgi**. Prá você ter uma idéia, o original pegava as informações no /etc/coyote/firewall, e o novo... também... hahaha.

Prá você poder visualizar bem isso sugiro que faça como eu: imprima os dois scripts e compare-os. Isso vai tornar tudo mais claro.

Figura 3

Esta tela (*figura 3*) apenas mostra o status de cada usuário perante o firewall e me dá a opção de liberar, bloquear ou mesmo excluir um usuário. As se posso excluir deveria também poder incluir, certo? Aí parti prá implementar isso. Difícil? Já ouviu aquela frase: "... nada se cria... tudo se transforma..." Pois é, continuei aproveitando o que já estava pronto. Só foi remodelar (*figura 4*). Mas foi um ótimo exercício que já me abriu um leque bem interessante.

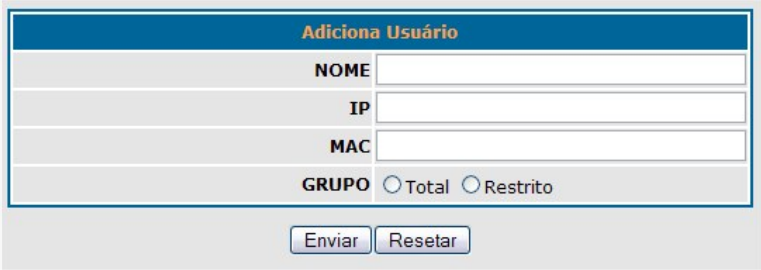


Figura 4

1.11 Aniversariantes

Aproveitando o ensejo, só prá mostrar mais um exemplo de como brinco de BFW.

Pesquisando na net sobre shell script, gosto de ficar vendo exemplos e deles tirar algumas idéias, parciais ou totais. Eis que encontro um que me chamou a atenção: envia email de aniversário. Dei uma olhada no código e pensei: acabo de conseguir fazer funcionar o add-on MAIL (que uso prá me enviar logs e backups) e assim poderia tirar mais um proveito dele.

Resolvi adaptá-lo mas estava no trabalho e lá eu não tinha como rodar o BFW. E tinha mais dez dias pela frente... o que fazer? Foi quando pentelhei o pessoal aqui prá me ajudar prá que eu colocasse prá rodar o BFW dentro do VMWare, usando até um "loop-back" (dispositivo sugerido pelo mestre Cláudio, já que eu queria faze-lo num notebook, com apenas uma placa de rede). Ali eu comecei a preparar esta adaptação.

Ficou pronta (*figura 5*), mas depois de eu ter feito o CGI citado anteriormente, para o controle de acesso, eu pensei: B-I-N-G-O!!! Vou implementar isso no WebAdmin. E aí já foi um pouco mais fácil. Agora, tenho inúmeros registros de aniversário (nome, data e email) e automaticamente, através de uma rodada diária (pelo cron), o BFW envia um email de felicitações para o(s) aniversariante(s) do dia (e outro prá mim mesmo, prá lembrar, afinal, tem algumas pessoas que a gente vai querer dar ainda um telefonema ou coisa e tal). E mostra ainda, no WebAdmin, os aniversariantes do mês.

Figura 5

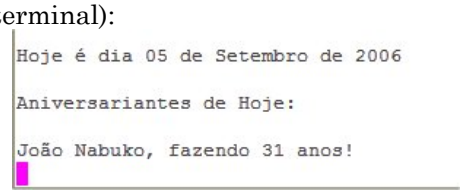
Cadastre (*figura 6*) quantas datas quiser (hehehe... tem limites, claro... mas é arquivo texto... levíssimo) e deixe o BFW cuidar de fazer a "média" por você. Legal, com um pouco de criatividade você pode usar esta AGENDA prá muitas outras coisas.

Figura 6

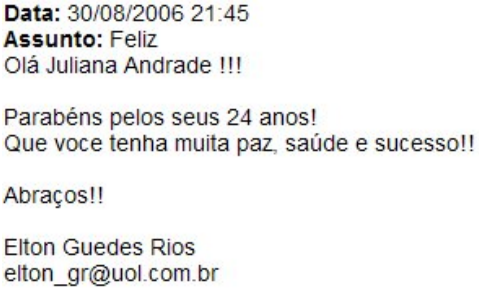
Veja o formato das linhas do arquivo:

```
# Agenda de Aniversário
07 Maio 1966 elton.gr@terra.com.br Elton Guedes Rios
16 Abril 1991 anninha_rios@hotmail.com Anna Paula Silva Rios
05 Outubro 1973 lourdesrios@uol.com.br Lourdes Maria de Oliveira
10 Outubro 1989 matheus.rios@hotmail.com Rillion de Oliveira Rios
15 Agosto 1966 lourdesrios@uol.com.br Manoel de Souza
10 Setembro 1975 jijellington@hotmail.com João Nabuko
28 Setembro 1956 m_nork@terra.com.br Marieta Norkiast
```

A mensagem ao administrador (via terminal):



E veja também um exemplo de email enviado por ele:



1.12 O que estou fazendo agora

Prá não perder o costume tenho algumas tarefas anotadas, das quais escolhi estas abaixo como minha próxima diversão:

- Criar uma forma de mexer no firewall pelo webadmin (opções originais) sem perder as informações adicionais que coloco nas linhas (para atender meus scripts);
- Sempre que for reiniciado o firewall trazer todos os usuários bloqueados (colocar opção no webadmin prá isso) pois hoje, se um "espertinho" desligar o servidor, por exemplo, alguns minutos antes de ele (usuário) ser bloqueado pelo cron, e só religar após o horário da regra, volta a última condição salva (no último backup) e aí ele pode ficar liberado direto, sacou?
- Colocar no Controle de acesso (WebAdmin) um campo entre MAC e Ações onde poderei entrar com um tempo (em minutos) que eu quero liberar o usuário. Se deixar em branco não haverá limite automático;
- Desabilitar "Libera" ou "Bloqueia" conforme o status. Exemplo: se o usuário aparece como "Bloqueado" então a opção para Bloqueio do mesmo fica desabilitada (cinza e sem ação);
- Permitir a liberação por grupos ao invés de só individualmente;