Reforçando a Segurança do FreeBSD 11

Sumário

O FreeBSD é por default, seguro. Mas podemos e devemos reforçar sua segurança defualt, especialmente quando nosso servidor instalou softwares de terceiros que podem trazer e geralmente trazem novas vulnerabilidades ao servidor como um todo.

Precisamos nos preocupar e cuidar de tudo que está relacionado ao servidor: o servidor em si, a hospedagem, o desktop usado pelo(s) administrador(es), cada um dos softwares de terceiros instalados e assim qualquer dos aspectos que possa comprometer a segurança. Cuidar da segurança é um verdadeiro casamento, algo que exige muito tempo e dedicação de que quem lida com ela.

Os cuidados com a segurança colaboram para que os sites e aplicativos sejam executados de forma esperada, rápida e sem interrumpção.

Princípios básicos de segurança:

- Hospede seu site em servidor seguro
- Efetue backup regularmente, especialmente antes de cada alteração no site
    A melhor opção atualmente para backup de sites com Joomla é o Akeeba Backup - https://www.akeebabackup.com/download.html
    Caso tenha dificuldade de usar o formato JPA, altere em Configuration - Archiver engine para ZIP format
    O Akeeba gera um backup com um instalador. Para restaurar apensa instale como se fosse instalar o Joomla
- Faça também backup dos scripts de configuração do servidor para o caso de uma reinstalação
- Lembre de fazer o backup do servidor com os recursos da hospedagem criando um snapshot ou backup
- Também faça teste de restore de vez em quando para garantir que o backup está íntegro
- A quantidade de cópias de backup a ser guardada depende da importância do site. Se mais importante mais cópias
- As cópias devem ser armazenadas em mídia confiável: HD e/ou DVD
- Efetue atualização com frequência. Mantenha o aviso de atualização do Joomla ativo para que receba um aviso por e-mail e atualize imediatamente
- Após a primeira atualização do servidor reinicie o mesmo
- Acessar de forma segura usando SSH (enxuto e configurado para acesso sem senha) e nunca via FTP
- Manter seu desktop seguro, usando um sistema operacional seguro no mesmo, com firewall e outros cuidados
- Use e abuse da comunidade com seus conhecimentos e generosidade para manter-se atualizado em termos de segurança e proteger seu site
- Use senhas fortes
- Use o SSL para proteger pelo menos o administrator e também a autenticação do Apache ou Nginx
- Use boas extensões para reforçar a segurança
- Remova todas as extensões que não estiver usando e não somente desabilite
- Evite instalar pacotes para desenvolvimento como gcc, make, etc e evite também instalar repositórios instáveis. Caso precise instalar remova imediatamente após
- Monitorar frequentemente os logs à procura de algo suspeito em todos os serviços ativos
- Use softwares tipo IDS que detectam intrusões
- Instalar um bom firewall de aplicativos como o mod_security

- Ficar bem atento, estudando, se informando sempre sobre segurança
- Logo após a configuração final do servidor já crie um backup ou snapshot do servidor e fique atento para criar outro logo que o servidor esteja concluído e bem configurado. Atualize este backup sempre após alterações do servidor

Planejando a política de segurança do Servidor

- Atualização e receber anúncios de vulnerabilidade
- Backup, snapshot e backup individual de cada site ou aplicativo
- SSL
- Authentication
- Auditoria
- Hardening para FreeBSD, para Apache, PHP, MySQL e Joomla
- Monitoramento
- Segurança para usuários
- Firewall
- Rede
- Permissões de arquivos
- Desabilitar serviços não usados
- Usar o Nível de segurança 3 após concluir a configuração do servidor
- Em Deus confiamos em tudo o restante autenticamos e monitoramos
- Nada de ftp, somente ssh com autenticação forte e em porta diferente da 22, preferir acima de 50000
- Ferramentas auxiliares: snort, aide, lynis, denyhosts, etc

O FreeBSD é bem seguro e conservador por default, mas mudar algumas configurações e reforçar alguns aspectos dos softwares/pacotes instalados é importante para manter o servidor mais resistente às tentativas de ataque, especialmente se o servidor estiver na internet.

Atualizações - as atualizações são muito importantes para manter o sistema operacional estável e seguro. Uma boa ideia é assinar a lista de anúncios de segurança do FreeBSD em https://lists.freebsd.org/mailman/listinfo/freebsd-announce

Lembre que após algumas atualizações precisamos reiniciar alguns serviços. Um exemplo é a atualização do php que precisa que o apache seja reiniciado para que seja usada. Outra é a atualização do kernel que precisa que o FreeBSD seja reiniciado para que seja usada.

Backup - Snapshot ou Backup do servidor para nosso servidor. Sempre que fizermos alterações ou mesmo atualizações do servidor é importante atualizar este backup.

IMPORTANTE - sempre antes de qualquer grande alteração nas configurações do servidor, devemos atualizar o backup ou snapshot, para em caso de problema poder ter de volta o backup do servidor funcionando.

Por padrão o FreeBSD é muito seguro. Problema de segurança no core do sistema é pouco frequente.
Usar boas práticas para tornar o FreeBSD mais resistente às ações maliciosas, especialmente em servidores publicanos na internet.

Aplicar hardening em um sistema operacional é alterar algumas das suas configurações default para que ele se torne mais seguro contra invasões.

Uma instalação padrão do FreeBSD é muito segura mas com as configurações abaixo procuramos reforçar esta segurança.

Ajustar usuários, mudando senha, impondo restrições como impedir login do root. A senha do root deve ser conhecida da menor quantidade de gente possível. Usar senhas bem fortes para todos os usuários. Contas de usuários devem ser auditadas frequentemente e mudadas ou excluídas as sem uso. O root nunca deve fazer login diretamente. Sempre logar como usuário comum e mudar apra sudo ou su quando precisar.

Hospedagem

A hospedagem, a empresa que administra o servidor que guarda os arquivos do site, é um dos itens mais importantes para a segurança do mesmo. Afinal de contas ele têm o controle sobre os servidores e assim pode implementar os mais diversos filtros de segurança que deixarão seu site em ambiente menos vulnerável ou o contrário, caso não implemente.

Antes de contratar colha a maior quantidade possível de informações sobre o serviço. Peça informações nos grupos que participa. Contate amigos que têm sites hospedados nele, questione sobre como é o serviço que recebem, suas características, para ver se atendem ao que planeja (sempre deixe uma folga em espaço, banda e quantidade de bancos). Existem muitas características sobre uma boa hospedagem que devemos considerar: espaço em disco, banda, quantidade de bancos, de domínios, suporte, suporte a conexões via SSH, segurança (vários itens relacionados), etc. Nunca contrate somente pelo menor preço. Pesquise antes.

Uma boa hospedagem tem um suporte rápido e competente, cuja equipe conhece bem suas funções e pode ajudar a resolver problemas e a detectá-los de forma ágil.
Uma boa hospedagem adota procedimentos que tornam seguro o seu site como:

Apache chrooted
PHP instalado como CGI
Apache modSecurity
Ativa a extensão Source Guardian no PHP

Escolha criteriosamente o servidor de hospedagem e o tipo. Tenha como um dos fatores de decisão a importância que o servidor dá à segurança.

Escolher um servidor para o site que seja da sua confiança, ou pelo menos que não desconfie dele.

Se possível evite servidores compartilhados.

Ao contratar pela primeira vez contrate num prazo mínimo para experimentar. E nunca esqueça de guardar um backup dos sites.

Caso use um VPS ou servidor dedicado fique atento à monitoração de ataques, TripWire e SAMHAIN são boas ferramentas para isso.
Cheque os logs regularmente.

Conexões ao servidor

Use somente conexões seguras ao conectar ao servidor. Se não conectar assim sua senha poderá ser interceptada. Evite ftp, ao invés use o gerenciador de arquivos o próprio cPanel. Caso exista, precisa conexão com SSH.

Configurações e Otimizações do SSH

Mudar para uma porta maior que 50.000 para maior segurança

nano /etc/ssh/sshd_config

Port 55522
LoginGraceTime 30
PermitRootLogin without-password
AllowUsers ribafs root

service sshd restart

Para que root acesse sem senha no SSH
PermitRootLogin without-password

Criar uma chave para o ssh no desktop

ssh-keygen -t rsa -b 4096

Para maior segurança usar a chave ED25519 ou ECDSA:

ssh-keygen -t ecdsa -b 4096

ssh-keygen -t ed25519 -b 4096

Então copie sua chave para o servidor com o comando abaixo, para que possa conectar sem digitar a senha. Na primeira vez te pedirá a senha mas sua senha do desktop, mas memorizará e não mais pedirá. Assim ficará mais seguro.

ssh-copy-id ribafs@ip_servidor -p 10522

Mesmo com scp não pedirá senha.

Sugestão - Criar um script para conectar:

sudo nano /usr/local/bin/servidor
ssh -p 55522 ribafs@128.199.63.251

sudo chmod +x /usr/local/bin/servidor

Conecte com
servidor

Atualização

A melhor defesa em termos de segurança é manter o sistema atualizado

pkg update
pkg upgrade

Após atualizar o kernel
shutdown -r now

Aplicando patches de segurança

freebsd-update fetch
freebsd-update install

Em caso de problema na atualização

freebsd-update rollback

Adicionar ao crontab

Para que seja feita de forma automática todos os dias

sudo suu
nano /etc/crontab

#Adicionei
@daily root freebsd-update -t freebsd cron

Receber Anúncios de Segurança por e-mail

Uma medida importante é se cadastrar na lista de anúncios de segurança do FreeBSD.

Esta aqui:
https://lists.freebsd.org/mailman/listinfo/freebsd-announce

Posts

sudo portsnap fetch extract
sudo portsnap fetch update

sudo pkg install portmaster

cd /usr/ports/ports-mgmt/portmaster
sudo make install clean

Checar por atualizações
portmaster -L

Evitando problemas na atualização

sudo nextboot -o "-s" -k kernel
sudo reboot

Backup

Atualização e backup são as medidas mais importantes e efetivas para garantir a segurança de um servidor.

Atualização e backup

Efetuar backup frequente do servidor

Efetuar Backups é uma das medidas mais importantes em termos de segurança, pois mesmo que o site tenha sido inteiramente destruído, se você tiver um backup confiável poderá em pouco tempo colocar seu site de volta.

Tenha um planejamento para backup e restore. Mantenha várias cópias do site (todos os arquivos e o banco) e faça testes locais de restauração.

Como você nunca sabe quando um ataque pode acontecer então faça backup e vários. Faça backup full do site com muita frequência: simplebackup e akeba backup ajudam (mas apenas para sites abaixo de 100MB, acima disso faça um backup manual).

Guarde não apenas uma cópia do backup, mas várias, pois pode ser que a última esteja comprometida ou corrompida.

Idealmente faça teste de restauração do site em micro local para se certificar logo após o backup.

Atualize o servidor com frequência.

Cuidados Iniciais

Usuários e Shell

Por questões de segurança é melhor usar o shell padrão do FreeBSD e na criação de usuários force senha forte e evite criar usuários sem necessidade.

Não esqueça dos seus clientes, eles podem ser uma fragilidade do servidor, portanto reforçe a importância de senhas fortes.

Use apenas ssh para conectar ao servidor. Em caso de sites, se precisar efetuar login use SSL e autenticação em áreas administrativas.

Para saber o que oo sitema está carregando na inicialização

grep YES /etc/defaults/rc.conf

grep YES /etc/rc.conf
ls /usr/local/etc/rc.d
/etc/inetd.conf

Delete serviços que não está usando
Mude as entradas "YES" no rc.conf para "NO"

Remova scripts de /usr/local/etc/rc.d

Comente serviços em
/etc/inetd.conf caso inetd esteja rodando

Cheque o crontab
Veja se não encontra alguma entrada estranha no crontab

Verificar se inetd está rodando com

service inetd onestatus

Senhas

Use senhas fortes para o cPanel, para usuários dos bancos, e-mails, para o Joomla, etc. De nada vai adiantar todo o cuidado com a segurança, se você deixa seu usuário com uma senha bem fácil ou medianamente fácil. A senha deve ser forte e senha forte é senha grande (8 ou maiis) e complexa (misturar letras maúsculas, minuúsculas, números e siímbolos).

Não use FTP, pois as senhas navegam em texto claro. Em seu lugar prefira sftp, FileZilla, scp, o gerenciador de arquivos do cPanel, etc.

Não dê oportunidade aos Crackers
Um cracker precisa ter duas coisas: oportunidade e habilidade. Vários crackers têm habilidade, não dê a eles a oportunidade.

Notícias Realistas

- Não existe segurança perfeita, portanto é melhor prevenir com backup
- Não existe a forma perfeita, sempre existem várias formas de se fazer, seja criativo e esforce-se
- Nada substitui a experiência, portanto estude e pratique. Aqui o que precisamos aprender: GNU/Linux, Apache, MySQL, SQL, PHP, HTTP, CSS, XML, RSS, TCP/IP, FTP, Subversion, JavaScript e Joomla, Linux, FreeBSD, etc.

Um sistema que é baseado no FreeBSD e que reforça a segurança é o HerdenedBSD - https://hardenedbsd.org/

Níveis de Segurança

Existem 5 níveis, -1, 0, 1, 2, 3.
O -1 é o mais baixo e o 3 o mais alto.

Para aprender sobre o sistema é recomendado o nível -1
Para um sistema em produção com segurança reforçada o nível 3. Lembrando que ao usar o nível 3 o firewall do host é imutável.
Nos níveis 1 e 2 não se pode ajustar os filtros de pacotes.

Em fases de manutenção configure como -1 e finalmente ajuste para 3:

sysctl kern.securelevel=3

Uma vez que você tenha apenas as portas de rede necessárias abertas, e você sabe quais programas estão usando essas portas, você sabe quais programas você deve estar mais preocupado em garantir. Se a equipe de segurança do FreeBSD enviar um anúncio de um problema com um serviço que você não executa, você pode seguramente atrasar a implementação de uma correção até a próxima manutenção. Se, no entanto, a equipe de segurança anuncia um buraco nos programas que você está usando, você sabe que tem que implementar uma correção o mais rápido possível. Se eles anunciam uma correção para um sério problema de segurança com um software de rede que você está usando, você sabe que deve ajir rapidamente. Simplesmente sendo capaz de responder de forma inteligente e rápida ao real riscos ajuda a protegê-lo contra a maioria dos intrusos.

Ferramentas como sinalizadores de arquivo e securelevels minimizam o dano que intrusos bem-sucedidos podem fazer. Finalmente, usando grupos para restringir seus próprios administradores de sistema a determinados seções do sistema podem proteger seus computadores de ambos os e danos deliberados.

Para checar o status do nível de segurança que o sistema está rodando

sysctl -n kern.securelevel

Existem 5 níveis, -1, 0, 1, 2, 3.
O -1 é o mais baixo e o 3 o mais alto.

-1 não tem nenhuma segurança implementada, destina-se apenas ao estudo.

Maior que 0 tem alguma segurança implementada.

Para aprender sobre o sistema é recomendado o nível -1
Para um sistema em produção com segurança reforçada o nível 3. Lembrando que ao usar o nívell 3 o firewall do host é imutável.

Nos níveis 1 e 2 não se pode ajustar os filtros de pacotes.

Em fases de manutenção configure como -1 e finalmente

Após tudo pronto ajuste para 3:

sysctl kern.securelevel=3

Para ficar permanente

```
kern_securelevel_enable="YES"
kern_securelevel="3"
```

Uma vez que você tenha apenas as portas de rede necessárias abertas, e você sabe quais programas estão usando essas portas, você sabe quais programas você deve estar mais preocupado em garantir.

Se a equipe de segurança do FreeBSD enviar um anúncio de um problema com um serviço que você não executa, você pode seguramente atrasar a implementação de uma correção até a próxima manutenção.

Se, no entanto, a equipe de segurança anuncia um buraco nos programas que você está usando, você sabe que tem que implementar uma correção o mais rápido possível.

Se eles anunciam uma correção para um sério problema de segurança com um software de rede que você está usando, você sabe que deve ajir rapidamente.

Simplesmente sendo capaz de responder de forma inteligente e rápida ao real risco ajuda a protegê-lo contra a maioria dos intrusos.

Ferramentas como sinalizadores de arquivo e securelevels minimizam o dano que intrusos bem-sucedidos podem fazer.

Finalmente, usando grupos para restringir seus próprios administradores de sistema a determinados seções do sistema podem proteger seus computadores de ambos os e danos deliberados.

Restringindo Permissões de Arquivos

Restrições de acesso somente para o usuário, o dono

Por padrão as permissões são assim:

ls -la /etc/sysctl.conf
-rw-r--r-- 1 root wheel

Com as configurações abaixo ficarão assim:

-rw-r----- 1 root wheel

Alguns arquivos importantes

chmod o= /etc/sysctl.conf
chmod o= /etc/ttys

```
chmod o= /etc/inetd.conf
chmod o= /etc/login.*
chmod o= /etc/fstab
chmod o= /etc/rc.conf
chmod o= /etc/ftpusers
chmod o= /etc/group
chmod o= /etc/host*
chmod o= /etc/inetd.conf
chmod o= /usr/bin/users
chmod o= /usr/bin/w
chmod o= /usr/bin/who
chmod o= /usr/bin/lastcomm
chmod o= /usr/bin/last
```

Atacantes tendem a limpar os logs quando finalizam seu ataque. Mudemos as permissões dos arquivos de log

```
chmod o= /var/log
chflags sappnd /var/log
chflags sappnd /var/log/*
```

Agora os logs não podem ser rodados

Criar o script

nano /usr/local/bin/perms

```
#!/bin/sh
clear;
echo "Aguarde enquanto configuro as permissões do /usr/local/www/apache24/data/$1";
echo "";
chown -R www:www /usr/local/www/apache24/data/$1;
find /usr/local/www/apache24/data/$1 -type d -exec chmod 755 {} \;
find /usr/local/www/apache24/data/$1 -type f -exec chmod 644 {} \;
echo "";
echo "Concluído!";
```

chmod +x /usr/local/bin/perms

Executar sempre que copiar arquivos ou descompactar em
/usr/local/www/apache24/data

Supondo que tenha criado a pasta
/usr/local/www/apache24/data/joomla
E tenha descompactado o joomla dentro dela então execute:

perms joomla

Harden executáveis

```
chflags -F schg /kernel
chflags -F schg /bin /sbin
```

Desabilitar cgi no httpd.conf

chmod 750 /usr/bin/perl
chmod 750 /usr/local/bin/perl

Remover Serviços Não Usados

Caso não use estes serviços abaixo:

pkg delete xinetd ypserv tftp-server telnet-server rsh-server

pkg list
pkg delete nome

Procurando portas escutando na rede (em seu desktop)

nmap -sT -O localhost

Procurar arquivos e diretórios com 777 e 666:

find /dir -xdev -type d \(-perm -0002 -a ! -perm -1000 \) -print

Procurar arquivos sem dono
find /dir -xdev \(-nouser -o -nogroup \) -print

Memória Compartilhada

Melhorar a segurança da memória compartilhada

Checar valores atuais
sysctl -a | grep -E "shmall|shmmax"

kern.ipc.shmall: 8192
kern.ipc.shmmax: 33554432

ou alternativamente:

ipcs -M

shminfo:
    shmmax: 33554432 (max shared memory segment size)
    shmmin: 1 (min shared memory segment size)
    shmmni: 192 (max number of shared memory identifiers)
    shmseg: 128 (max shared memory segments per process)
    shmall: 8192 (max amount of shared memory in pages)

Para fazer com que a configuração seja persistente, adicione as duas linhas para o :

nano /etc/sysctl.conf

kern.ipc.shmall=2097152
kern.ipc.shmmax=134217728

Descrição dos parâmetros:

kern.ipc.shmall: Número máximo de páginas (normalmente 4096 bytes) disponível para shared memory
kern.ipc.shmmax: Máximo tamanho do segmento da memória compartilhada em bytes

Para aplicar as alterações imediatamente em /etc/sysctl.conf, execute:

service sysctl restart

Impedir login direto do root

Para que somente usuários comuns possam fazer login no servidor, inclusive na console da hospedagem, edite o arquivo

Antes disso instale e use o sudo para fazer login como usuário comum e mudar para sudo quando precisar.

nano /etc/ttys

Faça um backup full do sistema ou atualize o snapshot

E troque todas as ocorrências de 'secure' por 'insecure'

Reforçando a Segurança dos usuários

Cada usuário deve ter sua conta restrita
Devem usar senhas fortes
A senha do root deve ser conhecida pela menor quantidade de pessoas possível
As contas de usuários devem ser auditadas periodicamente e desabilitar ou remover usuários que não são mais usados

Comando sudo do linux

É uma boa forma de ganhar acesso de root temporariamente. Não vem instalado por default no FreeBSD 11, mas pode ser instalado com

pkg install sudo

Pelos ports
cd /usr/ports/security/sudo
make install

Adicionar um usuário ao sudoers ou para um grupo que já esteja lá, como o wheel

nano /usr/local/etc/sudoers

ribafs ALL=(ALL)   ALL

ou adicionar ao grupo wheel, que é um grupo com privilégios administrativos

pw usermod ribafs -G wheel

Reforçando

Meu exemplo

```
hostname="ribafs"
cloudinit_enable="YES"
sshd_enable="YES"
digitaloceanpre="YES"
digitalocean="YES"
zfs_enable="YES"

ntpd_enable="YES"
ntpd_sync_on_start="YES"

clear_tmp_enable="YES"
syslog_flags="-ss"
sendmail_enable="NONE"
sendmail_submit_enable="NO"
sendmail_outbount_enable="NO"
sendmail_msp_queue_enable="NO"
sendmail_cert_create="NO"
kern_securelevel_enable="YES
indet_enable="NO"
icmp_drop_redirect="YES"
tcp_drop_synfin="YES"
icmp_log_redirect="YES"
portmap_enable="NO"
log_in_vain="YES"

sshguard_enable="YES"

apache24_enable="yes"
mysql_enable="yes"

#Adicionei
firewall_enable="YES"
firewall_quiet="YES"
firewall_type="workstation"
firewall_myservices="65522 80 443"
firewall_allowservices="any"
firewall_logdeny="YES"
#firewall_script="/usr/local/etc/IPFW.rules"
```

```
# DigitalOcean Dynamic Configuration lines and the immediate line below it,
# are removed each boot.

# DigitalOcean Dynamic Configuration
defaultrouter="167.99.160.1"
# DigitalOcean Dynamic Configuration
ipv6_defaultrouter=""
# DigitalOcean Dynamic Configuration
ipv6_activate_all_interfaces="yes"
```

sudo nano /etc/sysctl.conf

Adicionar ou mudar

```
security.bsd.see_other_uids=0
security.bsd.see_other_gids=0
security.bsd.unprivileged_read_msgbuf=0
security.bsd.unprivileged_proc_debug=0
security.bsd.stack_guard_page=1
kern.randompid=$(jot -r 1 9999)
kern.securelevel=3
kern.ps.showallprocs=0
net.inet.tcp.blackhole=2
net.inet.udp.blackhole=1
net.inet.ip.random_id=1

net.inet.tcp.sendbuf_max=16777216
net.inet.tcp.recvbuf_max=16777216
net.inet.tcp.sendbuf_auto=1
net.inet.tcp.recvbuf_auto=1
net.inet.tcp.sendbuf_inc=16384
net.inet.tcp.recvbuf_inc=524288

net.inet.tcp.mssdflt=1460
net.inet.tcp.minmss=1300
net.inet.tcp.syncache.rexmtlimit=0
net.inet.tcp.syncookies=0
net.inet.tcp.tso=0
net.inet.ip.check_interface=1
net.inet.ip.process_options=0
net.inet.ip.redirect=0
net.inet.icmp.drop_redirect=1
net.inet.tcp.always_keepalive=0
net.inet.tcp.drop_synfin=1
net.inet.tcp.ecn.enable=1
net.inet.tcp.icmp_may_rst=0
net.inet.tcp.msl=5000
net.inet.carp.preempt=1
```

e estas linhas para /etc/rc.conf:

```
clear_tmp_enable="YES"
```

```
syslog_flags="-ss"
sendmail_enable="NONE"
sendmail_submit_enable="NO"
sendmail_outbount_enable="NO"
sendmail_msp_queue_enable="NO"
sendmail_cert_create="NO"
kern_securelevel_enable="YES
indet_enable="NO"
icmp_drop_redirect="YES"
tcp_drop_synfin="YES"
icmp_log_redirect="YES"
portmap_enable="NO"
log_in_vain="YES"
```

Se usar o Firewall PF

```
pf_enable="YES"
pf_rules="/etc/firewall"
pf_flags=""
pflog_enable="YES"
pflog_logfile="/var/log/pflog"
pflog_flags=""
```

Digital Ocean adicionou

```
# DigitalOcean Dynamic Configuration
defaultrouter="167.99.160.1"
# DigitalOcean Dynamic Configuration
ifconfig_vtnet0="inet 167.99.175.44 netmask 255.255.240.0"
# DigitalOcean Dynamic Configuration
ifconfig_vtnet0_alias0="inet 10.46.0.5 netmask 255.255.0.0"
# DigitalOcean Dynamic Configuration
ipv6_defaultrouter=""
# DigitalOcean Dynamic Configuration
ipv6_activate_all_interfaces="yes"
```

Ao instalar o FreeBSD ele cria dois diretórios temporários: /tmp e /var/tmp. Vamos excluir o último e criar um link simbólico de /tmp para ele

```
sudo su

mv /var/tmp/* /tmp/
rm -rf /var/tmp
ln -s /tmp /var/tmp
```

Timezone

```
sudo tzsetup
```

NTP

sudo nano /etc/rc.conf

ntpd_enable="YES"
ntpd_sync_on_start="YES"

sudo service ntpd restart

Configurações
/etc/ntp.conf

Checando vulnerabilidade conhecidas de softwares

sudo pkg audit -F

Configurações

Base do sistema em
/etc

E atualziado com
freebsd-update

Softwares opcionais em
/usr/local/etc

Ports em
/usr/ports

Atualizados com:
sudo portsnap fetch update

Ao procurar por um software opcional e somente o encontrar na pasta dos ports, significa que ele não está instalado.

Monitorando/Auditando - AIDE

pkg install aide

via ports
make install /usr/ports/security/aide

sudo su
pkg install aide

ou com pors
make install /usr/ports/security/aide

aide --help

aide -v

aide --init

cd /var/lib/aide
ou
cd /var/db/aide/database

ls -lt

mv /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz

ls -lt

aide --check

Para comparar os bancos de dados execute

aide

Criar arquivo de teste
touch /usr/sbin/mytestfile.txt

aide --check

aide --update

ls -lt

mv aide.db.gz aide.db.gz-Marc152018`

mv aide.db.new.gz aide.db.gz

Automatizar

crontab -e

06 01 * * 0-6 /var/log/aide/chkaide.sh

cat /var/log/aide/chaide.sh

Configurações
/etc/aide.conf

Crédito
https://www.digitalocean.com/community/tutorials/how-to-install-aide-on-a-digitalocean-vps

LYNIS

Este software nasceu no FreeBSD e atualmente também é utilizado nos Linux. É um software open surce para auditoria do sistema.

Instalação

Via ports
cd /usr/ports/security/lynis
make install clean

Via PKG
pkg install security/lynis

Executar
lynis -c -Q

-c - executa scan com todos os testes habilitados
-Q - evitar aguardar por feedback do usuário

Logs
/var/log/lynis.log

nano /etc/periodic.conf

daily_clean_tmps_enable="YES"
daily_clean_hoststat_enable="NO"
daily_status_include_submit_mailq="NO"
daily_status_mail_rejects_enable="NO"
daily_queuerun_enable="NO"
daily_submit_queuerun="NO"

Firewall com PF

O Packet Filter é bem simples de configurar. Somente habilite se deseja acesso externo ao servidor.

O PF já vem embutido no kernel do FreeBSD, portanto não requer instalação de nenhum pacote.

Configurações e Regras
nano /etc/pf.conf

Testar regras
pfctl -nf /etc/pf.conf

Aplicar regras
pfctl -f /etc/pf.conf

Habilitar
pfctl -e

Exemplo com regras simples:

nano /etc/pf.conf

Início

ext_if="bge0"

IP_FREEBSD_HOST="192.168.0.xxx"
IP_WEB="192.168.0.xxx"
SSH_HOSTS= "{" $IP_FREEBSD_HOST $IP_WEB "}"
ICMP_TYPES="{echoreq,unreach}"

PORT_WEB="{80,443}"
PORT_SSH="{22,1413}"
PORT_ZABBIX="{10059}"

scrub in all
set skip on lo0
block in on vtnet0
pass in on vtnet0 proto icmp
pass in on vtnet0 proto icmp6
pass in on vtnet0 proto tcp from any to any port 65522
pass in on vtnet0 proto tcp from any to any port 80
pass in on vtnet0 proto tcp from any to any port 443

pass quick proto tcp from 177.x.x.x to vtnet0 port 65522 flags S/SA keep state
pass in quick proto tcp from <workssh> to $SSH_HOSTS port $PORT_SSH flags S/SA keep state (max-src-conn 10, max-src-conn-rate 3/5, overload <fail2ban> flush)

Final

pfctl -f /etc/pf.conf

service pf reload

reboot

Logs
/var/log/pflog

Criar o arquivo e adicionar os IPs confiáveis

nano /etc/trusted

Hosting

```
1.2.0.0/16

# My friends
1.2.4.0/24
```

Outro Exemplo

```
#These limits are far beyond FreeBSD's pf default limit.
set limit { states 100000, frags 25000, src-nodes 50000 }

pass quick on lo0 all

#The default Vulture's pf policy is:
# - Drop and log everything in input
# - Accept any outgoing traffic
# - IPV6 is enabled by default

block in log all
pass in proto icmp6 all
pass out proto icmp6 all
pass out all keep state

# Whitelist for Vulture Cluster
# This table is auto managed by Vulture
table <vulture_cluster> persist file "/usr/local/etc/pf.vulturecluster.conf"
pass in quick from <vulture_cluster>

# Brute force / SYN Flood / DDOS mitigation rule
# Use pfctl -t abusive_hosts -T show to show currently blacklisted hosts
# You can add manual persistent IP in this file
table <abusive_hosts> persist file "/usr/local/etc/pf.abuse.conf"
block in log quick from <abusive_hosts>

# SSH brute-force protection
# Use pfctl -t ssh_abusive_hosts -T show to show currently blacklisted hosts for ssh port
# You can add manual persistent IP in this file
table <ssh_abusive_hosts> persist file "/usr/local/etc/pf.sshabuse.conf"
block in log quick from <ssh_abusive_hosts>

# Incoming policy: By default, HTTP and HTTPS ports are accepted from everywhere on
em0
pass log quick inet proto tcp from any to em0 port { 80, 443 } flags S/SA keep state \
                        (max-src-conn 100, max-src-conn-rate 15/5, \
                         overload <abusive_hosts> flush global)

# Incoming policy: By default, SSH port is accepted from everywhere
pass log quick inet proto tcp from any to em0 port 22 flags S/SA keep state \
                        (max-src-conn 100, max-src-conn-rate 3/5, \
                         overload <ssh_abusive_hosts> flush global)

# Incoming policy: By default inter-cluster communication are allowed from everywhere on
em0
pass quick proto tcp from any to em0 port 8000 flags S/SA keep state
```

pass quick proto tcp from any to em0 port { 6379, 9091, 26379 } flags S/SA keep state

---- Allow CARP communications between nodes
pass in proto carp all

Ver quem está tentando conectar para nosso servidor
tcpdump -n -e -ttt -i pflog0

Mostrar histórico
tcpdump -n -e -ttt -r /var/log/pflog

Tentativas de força bruta
pfctl -t bruteforcers -T show

Monitorar segurança das atualizações dos ports instalados

cd /usr/ports/ports-mgmt/portaudit
usr/ports/ports-mgmt/portaudir
make install clean

Verificar que portas tcp e udp estão abertas

Portas IPv4
sockstat -4l

Portas IPv6
sockstat -6l

Ambos
sockstat -l

Reforçar Chave pública do SSH

Preferir acesso com chaves do SSH do que com senha

Temos diversas opções para gerar a chave, da menos para a mais segura:

ssh-keygen -t rsa -b 4096
ssh-keygen -t dsa
ssh-keygen -t ecdsa -b 521
ssh-keygen -t ed25519

Após gerar a chave enviar para o servidor, do desktop assim:

ssh-copy-id -i ribafs@ribafs.org -p 65522

ssh-copy-id -i ~/.ssh/tatu-key-ecdsa user@host

Usando Denyhosts para bloquear tentativas falhas de login

Scannear logs e banir hosts suspeitos

Denyhosts – bloqueia ataques de SSH adicionando entradas ao /etc/hosts.dny. Também avisa ao administrador sobre hosts suspeitos, ataques de usuários e logins suspeitos.

pkg install denyhosts

E referir para ele em:

/etc/hosts.deny

nano /usr/local/etc/denyhosts.conf

Adicionar a linha
SSHD_PORT = 65522

Descomentar a linha
BLOCK_SERVICE = sshd

Habilitar

nano /etc/rc.conf

denyhosts_enable="YES"

service denyhosts start

nano /etc/rc.conf

denyhosts_enable=YES

Após instalar edite o
nano /usr/local/etc/denyhosts.conf

E atualize seu e-mail e outras configurações que desejar.

ADMIN_EMAIL = ribafs@gmail.com
SMTP_HOST = localhost
SMTP_PORT = 25
#SMTP_USERNAME=foo
#SMTP_PASSWORD=bar
SMTP_FROM = DenyHosts nobody@localhost
#SYSLOG_REPORT=YES
service denyhosts restart

Criar o hosts.allow
nano /etc/hosts.allow

ftpd : all : deny

#Permitindo somente casa e trabalho
sshd : 177.130.208.59 187.130.18.59 : deny

ALL : 216.136.204.0/255.255.255.0 : deny

Exemplo

#reject all connections from hosts with invalid DNS and from our competitor
ALL : PARANOID 10.5.4.0/23 : deny
#localhost can talk to itself
ALL : localhost : allow
#our local network may access portmap, but no others
portmap : ALL EXCEPT 192.168.0.0/16 : deny
#allow SSH, pop3, and ftp, deny everything else
sshd, POP3, ftpd : ALL : allow
ALL : ALL : deny

service denyhosts start

Senhas Fortes

De que vai adiantar ter todo este trabalho de escolher uma boa hospedagem, de isntalar um sistema operacional seguro, atualizar o sistema e efetuar diversas medidas para melhorar a segurança, nada vai adiantar se usarmos senhas fracas.

É como cagar e não limpar o c*.

Senhas fortes são grandes (8 dígitos ou mais) e usam uma mistura de algarismos, letras minúsculas, letras maiúsculas e símbolos.

Pequena página em JavaScript que critica a força de senhas

```
<script>
 function TestaSenha(valor) {
    var d = document.getElementById('seguranca');
    // Expressões Regulares
    ERaz = /[a-z]/;
    ERAZ = /[A-Z]/;
    ER09 = /[0-9]/;
    ERxx = /[@!#$%&*+=?|-]/;
    // Teste da String
    if(valor.length == ''){
      d.innerHTML = '<b>Segurança da senha: !</b>';
    } else {
      if(valor.length < 5){
        d.innerHTML = '<b>Segurança da senha: <font color=\'red\'> BAIXA</font></b>';
      } else {
```

```
      if(valor.length > 7 && valor.search(ERaz) != -1 && valor.search(ERAZ) != -1 &&
valor.search(ER09) != -1 || valor.search(ERaz) != -1 && valor.search(ERAZ) != -1 &&
valor.search(ERxx) || valor.search(ERaz) != -1 && valor.search(ERxx) != -1 &&
valor.search(ER09) || valor.search(ERxx) != -1 && valor.search(ERAZ) != -1 &&
valor.search(ER09)){
        d.innerHTML = '<b>Segurança da senha: <font color=\'green\'> ALTA</font></b>';
       } else {
        if(valor.search(ERaz) != -1 && valor.search(ERAZ) != -1 || valor.search(ERaz) != -1
&& valor.search(ER09) != -1 || valor.search(ERaz) != -1 && valor.search(ERxx) != -1 ||
valor.search(ERAZ) != -1 && valor.search(ER09) != -1 ||valor.search(ERAZ) != -1 &&
valor.search(ERxx) != -1 ||valor.search(ER09) != -1 && valor.search(ERxx) != -1){
          d.innerHTML = '<b>Segurança da senha: <font color=\'orange\'>
MEDIA</font></b>';
        } else {
          d.innerHTML = '<b>Segurança da senha: <font color=\'red\'> BAIXA</font></b>';
        }
       }
     }
   }
 }
</script>

<body>
<h2>Teste de Segurança de Senha (JavaScript)</h2>
<form name=frm>
Login &amp;nbsp;<input name="login"><br>
Senha <input type=password name=senha onKeyPress="TestaSenha(senha.value)"><div
id="seguranca"></div><br>
<input type=submit onClick="TestaSenha(senha.value)" value="Acessar">
</form>
<pre>
```

Teste de Segurança da senha em JavaScript

Autor: André Lourenço Pedroso

Alguns de vocês devem ter visto no Hotmail(tm), por exemplo, um recuro onde
é feito um teste da senha, mostrando o seu nivél de segurança.

Para aqueles que acharam esse recurso interessante, mostro nesse pequeno
artigo um exemplo em JavaScript.

Os testes seguem a seguinte lógica:

- Baixa segurança - Senha que contem um tipo de caracter.
- Média segurança - Senha que tenha mais de quatro digitos e contenha no mínimo dois tipos de caracteres.
- Alta segurança - Senha que tenha mais de sete digitos e contenha no mínimo três tipos de caracteres diferentes.

Dica recebida da Dicas-L (http://www.dicas-l.com.br).

</pre>

</body>

Criptografia

OpenSSL

Criptografar um arquivo texto ou tar. O arquivo é criptografado com uma senha e quem a conhecer poderá descriprografar

Criptografar o arquivo.txt para arquivo.aes

openssl aes-128-cbc -salt -in arquivo.txt -out arquivo.aes

Descriptografar o arquivo.aes para arquivo.txt
openssl aes-128-cbc -d -salt -in arquivo.aes -out arquivo.txt

Empacotar com tar e criptografar todo um diretório

tar -cf - pasta | openssl aes-128-cbc -salt -out pasta.tar.aes

Desempacotar e descriptografar

openssl aes-128-cbc -d -salt -in pasta.tar.aes | tar -x -f -

Empacotar e zipar todo um diretório criptografar

tar -zcf - pasta | openssl aes-128-cbc -salt -out pasta.tar.gz.aes

Deszipar e descriptografar

openssl aes-128-cbc -d -salt -in pasta.tar.gz.aes | tar -xz -f -

Use -k senha após aes-128-cbc para evitar a requisição interativa da senha
Use aes-256-cbc ao invés de aes-128-cbc para ter uma criptografia mais forte, mas exigirá mais CPU

GPG

O arquivo é criptografado com uma senha e quem a conhecer poderá descriprografar

GPG adiciona uma extensão .gpg ao arquivo criptografado

Criptografar o arquivo file
gpg -c file

Descriptografar
gpg file.gpg

Usando chaves
gpg --gen-key

Isso pode demorar

~/.gnupg/pubring.gpg # Contains your public keys and all others imported
~/.gnupg/secring.gpg # Can contain more than one private key

Opções mais usadas

-e encrypt data
-d decrypt data
-r NAME encrypt for recipient NAME (or 'Full Name' or 'email@domain')
-a create ascii armored output of a key
-o use as output file

Criptografia apenas para uso pessoal

gpg -e -r 'Your Name' file # Encrypt with your public key
gpg -o file -d file.gpg # Decrypt. Use -o or it goes to stdout

Criptografar e Descriptografar com chave

gpg -a -o alicekey.asc --export 'Alice' # Alice exported her key in ascii file.
gpg --send-keys --keyserver subkeys.pgp.net KEYID # Alice put her key on a server.
gpg --import alicekey.asc # You import her key into your pubring.
gpg --search-keys --keyserver subkeys.pgp.net 'Alice' # or get her key from a server.

Logo que a chave é importada fica fácil de criptografar e descriptografar

gpg -e -r 'Alice' file # Encrypt the file for Alice.
gpg -d file.gpg -o file # Decrypt a file encrypted by Alice for you.

Administração de chaves

Key administration

gpg --list-keys # list public keys and see the KEYIDS
The KEYID follows the '/' e.g. for: pub 1024D/D12B77CE the KEYID is D12B77CE
gpg --gen-revoke 'Your Name' # generate revocation certificate
gpg --list-secret-keys # list private keys
gpg --delete-keys NAME # delete a public key from local key ring
gpg --delete-secret-key NAME # delete a secret key from local key ring
gpg --fingerprint KEYID # Show the fingerprint of the key
gpg --edit-key KEYID # Edit key (e.g sign or add/del email)

Fail2Ban

Fail2Ban

O fail2ban deve ser instalado após a instalação do AMP/EMP

O fail2ban é mais eficiente que o denyhosts, pois ele extende a monitoração de logs para outros serviços além do ssh, como o apache, courier, ftp e mais.

O fail2ban escaneia arquivos de log e bane IPs que parecem suspeitos (muitas tentativas erradas de senha, procurando por exploits, etc)
Geralmente bloqueia através do firewall por um certo tempo que é configurável

Instalação

É sempre bom verificar o que existe antes de instalar, assim como atualizar o pkg

pkg search fail2ban

pkg update

pkg install fail2ban

Após instalar edite

nano /usr/local/etc/fail2ban/jail.conf

E crie o filtro de regras requerido
Ative todos os serviços que deseja que o fail2ban monitore

Para que monitore o ssh, altere ou adicione
enable = true

OBS: atente para mudar de ssh para o número que escolheu, caso não use a 22.

[sshd]

enabled = true
port   = ssh
filter   = sshd
logpath = /var/log/auth.log
maxretry = 3
Caso o seu ssh esteja usando outra porta, mude port = suaporta

Checar status:

nano /etc/rc.conf

fail2ban_enable=”YES”

service fail2ban start

fail2ban-client status

Whitelisting

Whitelisting é configurada no jail.conf usando uma lista separada por espaço

```
[DEFAULT]
# "ignoreip" can be an IP address, a CIDR mask or a DNS host. Fail2ban will not
# ban a host which matches an address in this list. Several addresses can be
# defined using space separator.
Ignoreip = 127.0.0.1 192.168.1.0/24 8.8.8.8
```

ModSecurity

Instalação e configurações do mod_secutiry no FreeBSD 11 com Apache 2.4

O mod_security é um firewall de aplicativos web (WAF), que pode bloquear ataques genéricos e específicos.

Instalar

pkg install ap24-mod_security

Aparece:

Message from ap24-mod_security-2.9.2_2:

You have installed ModSecurity.
To enable ModSecurity in Apache, follow the instructions in

/usr/local/etc/apache24/modules.d/280_mod_security.conf

Most users will use the signatures from the OWASP Core Rule Set (CRS).
For configuration instructions, see /usr/local/share/doc/mod_security2/README.

Habilitar mod_security no Apache 2.4

nano /usr/local/etc/apache24/modules.d/280_mod_security.conf

```
## module file for mod_security
##
## PROVIDE: mod_security2
## REQUIRE: mod_unique_id

##
## To enable ModSecurity in Apache, enable the modules
##  mod_unique_id (in httpd.conf) and
##  mod_security2 in this config file
##
## Additionally, load configuration and rules with an Include line from
##  /usr/local/etc/modsecurity/*.conf
##
## Most users will use the signatures from the OWASP Core Rule Set (CRS).
```

```
## For configuration instructions, see /usr/local/share/doc/mod_security2/README.
##

## apache modules for mod_security
LoadModule unique_id_module libexec/apache24/mod_unique_id.so
LoadModule security2_module libexec/apache24/mod_security2.so
Include /usr/local/etc/modsecurity/*.conf
```

```
apachectl restart
```

Verificar se mod_security está carregado

```
tail -f /var/log/httpd-error.log
```

Recebendo o conjunto de regras do core

```
pkg install git
cd /usr/local/etc/modsecurity
git clone https://github.com/SpiderLabs/owasp-modsecurity-crs

cp owasp-modsecurity-crs/crs-setup.conf.example crs.conf

nano /usr/local/etc/apache24/modules.d/280_mod_security.conf

vim: set filetype=apache:
##
## module file for mod_security
##
## PROVIDE: mod_security2
## REQUIRE: mod_unique_id

##
## To enable ModSecurity in Apache, enable the modules
##  mod_unique_id (in httpd.conf) and
##  mod_security2 in this config file
##
## Additionally, load configuration and rules with an Include line from
##  /usr/local/etc/modsecurity/*.conf
##
## Most users will use the signatures from the OWASP Core Rule Set (CRS).
## For configuration instructions, see /usr/local/share/doc/mod_security2/README.
##

## apache modules for mod_security
LoadModule unique_id_module libexec/apache24/mod_unique_id.so
LoadModule security2_module libexec/apache24/mod_security2.so
Include /usr/local/etc/modsecurity/*.conf
Include /usr/local/etc/modsecurity/owasp-modsecurity-crs/rules/*.conf
```

Configurar para bloquear

nano /usr/local/etc/modsecurity/modsecurity.conf

#SecRuleEngine DetectionOnly
SecRuleEngine On

apachectl restart

Log Rotation

mkdir /usr/local/etc/newsyslog.conf.d

touch /usr/local/etc/newsyslog.conf.d/httpdlog.conf

nano /usr/local/etc/newsyslog.conf.d/httpdlog.conf

Adicionar

```
/var/log/httpd-access.log          600  7     *    @T12   B    /var/run/httpd.pid 30
/var/log/httpd-error.log           600  7     *    @T12   B    /var/run/httpd.pid 30
/var/log/httpd-ssl_request.log     600  7     *    @T12   B    /var/run/httpd.pid 30
/var/log/modsec_audit.log          600  7     *    @T12   B    /var/run/httpd.pid 30
```

service newsyslog restart

Atualizar o CRS

cd /usr/local/etc/modsecurity/owasp-modsecurity-crs
git pull
apachectl restart

Referência
https://loga.us/2017/02/22/freebsd-apache-modsecurity/

Testando segurança de sites

Chame:
http://ribafs.org/index.php?secret_file=/etc/passwd

Será barrado com o aviso:
Forbidden
You don't have permission to access / on this server.

Simulação básica de ataque de SQL Injection
http://ribafs.org/?id=23' or '1'='1

Também barrado.

Ao efetuar os testes acima em servidor local eles não foram recusados,mas o segundo recebeu 404 de um site em Joomla com Helix.

ModEvasiive

pkg search evasive

pkg install ap24-mod_evasive

[preparing module `evasive20' in /usr/local/etc/apache24/httpd.conf]

mkdir /var/log/mod_evasive
chown www:www /var/log/mod_evasive/

Criar
nano /usr/local/etc/apache24/modules.d/mod-evasive.conf

Adicione:

<ifmodule mod_evasive20.c>
DOSHashTableSize 3097
DOSPageCount 2
DOSSiteCount 50
DOSPageInterval 1
DOSSiteInterval 1
DOSBlockingPeriod 10
DOSLogDir /var/log/mod_evasive
DOSEmailNotify ribafs@gmail.com
DOSWhitelist 127.0.0.1
DOSWhitelist 177.130.202.171
</ifmodule>

Lembrando que o IP 177… é o IP que será liberado, o do meu desktop.

service apache24 restart

Logs
tail /var/log/httpd-access.log

Para adicionar um IP na white list adicione uma linha no arquivo acima:
DOSWhitelist 162.13.23.127

Devemos adicionar o IP de cada computador que usamos para administrar o servidor.

É bom lembrar que softwares como o mod_evasive e o denyhosts precisam ter em sua whitelist nossos IP de acesso, caso contrário teremos problema. Ainda bem que o Ocean tem uma console via web.

Com isso quando ele considerar algo que mereça mandará para a blacklist e te enviará um e-mail

Para não mais receber os e-mails mude para DOSSystemCommand ao invés de DOSEmailNotify, assim:

nano /usr/local/etc/apache24/modules.d/mod-evasive.conf

DOSSystemCommand "echo 'mod_evasive HTTP Blacklisted %s more info here: www.projecthoneypot.org/ip_%s' | mail -s 'Blocked IP by mod_evasive' root@localhost"

SSHGuard

Reforçar a segurança do SSH no FreeBSD com SSHGUARD

Após a instalação e configuração com sucesso do PF

pkg install sshguard

2.0.0_1

Após a instalação mostra:

```
  Sshguard installed successfully.

  You can start sshguard as a daemon by using the
  rc.d script installed at /usr/local/etc/rc.d/sshguard .

  See sshguard-setup(7) and http://www.sshguard.net/docs/setup for additional info.

  Please note that a few rc script parameters have been renamed to better reflect the
documentation:

  sshguard_safety_thresh -> sshguard_danger_thresh
  sshguard_pardon_min_interval -> sshguard_release_interval
  sshguard_prescribe_interval -> sshguard_reset_interval
```

nano /etc/rc.conf

Adicione

```
# sshguard
sshguard_enable="YES"
sshguard_danger_thresh="30"
sshguard_release_interval="600"
sshguard_reset_interval="7200"
```

nano /etc/pf.conf

```
table <sshguard> persist
#...
```

block in quick on bge0 from <sshguard> label "ssh bruteforce"

A ordem das linhas acima é importante, se alterada pode causar erro no pf

service sshguard start

pfctl -f /etc/pf.conf

service pf start

Após o comando acima travou meu terminal e saiu com:
packet_write_wait: Connection to 46.101.50.99 port 65522: Broken pipe

Então fui até a hospedagem e acessei pela console

Ao efetuar o acesso pela console aparece a mensagem de que tenho e-mail

Digito mail e teclo enter
Aparecem 3. Digito 1 para ler o security

Ele me avisa para checar o setuid de arquivos e dispositivos em:
var/log/setuid.today

Também para checar permissões negativas de grupos em:
/var/log/mount.today

Checar os uid 0
root 0
toor 0

Checar contas sem senha:
/etc/login.conf

Depois de ler estes e-mails eu carreguei as regras do pf e o reiniciei

Voltei ao meu desktop e acessei o servidor sem problema

Testando

pgrep -lfa ssh

pfctl -t sshguard -T show

pfctl -sr

Questões
https://forums.freebsd.org/threads/howto-set-up-and-configure-security-sshguard-pf.39196/

https://blog.mwzhang.com/2016/02/20/secure-ssh-using-sshguard-and-pf-in-freebsd/
https://gist.github.com/WillSquire/b0546bb8ab901f16555aba2e953767d9

Protegendo o Administrator com a Autenticação do Apache

Ou para proteger outro diretório qualquer do servidor.

Protegendo diretório com a autenticação do Apache 2.4 no FreeBSD 11.1

htpasswd -c /usr/local/etc/apache24/.httpd_access ribafs

cat /usr/local/etc/apache24/.httpd_access

Criar um directório protegido

Adicionar

nano /usr/local/etc/apache24/httpd.conf

Abaixo do bloco <Directory> existente

```
<Directory "/usr/local/www/apache24/data/administrator">
    AuthType Basic
    AuthName "Restricted Content"
    IndexIgnore .*
    AuthName protectthis
    AuthUserFile  /usr/local/etc/apache24/.httpd_access
    AuthGroupFile /usr/local/etc/apache24/.httpd_access_group
    AuthType Basic
    <Limit GET>
       # A linha abaixo é para um grupo
       #Require valid-user
       require user ribafs
    </Limit>
</Directory>
```

mkdir /usr/local/www/apache24/data/administrator

service apache24 restart

Testando

https://ribafs.org/administrator/

Caso o DNS não tenha propagado ainda use o iP.

Usando o SSL para Criptografar sites

Se o site usa login e precisa autenticar usuários é importante que se instale o SSL para criptografar as senhas. Sempre que alguém fizer login no site sua senha será criptografada antes de ser enviada para o servidor.

Instalação do SSL no Apache 2.4 do FreeBSD 11.1

Criar o certificado:

Criar o script abaixo

nano ssl.sh

```
#!/bin/sh
mkdir -p /root/mycert
cd /root/mycert
mkdir -p /usr/local/etc/apache24/ssl.key
mkdir -p /usr/local/etc/apache24/ssl.crt
chmod 0400 /usr/local/etc/apache24/ssl.key
chmod 0400 /usr/local/etc/apache24/ssl.crt
openssl genrsa -des3 -out $1.key 1024
openssl req -new -x509 -nodes -sha256 -days 365 -key $1.key -out $1.crt
cp $1.key $1.key.orig
openssl rsa -in $1.key.orig -out $1.key
cp $1.key /usr/local/etc/apache24/ssl.key/
cp $1.crt /usr/local/etc/apache24/ssl.crt/
chmod 0400 /usr/local/etc/apache24/ssl.key/$1.key
chmod 0400 /usr/local/etc/apache24/ssl.crt/$1.crt
### Final

/usr/local/etc/apache24/ssl.crt/ribafs
chmod +x ssl.sh
```

Executar:

./ssl.sh ribafs

nano /usr/local/etc/apache24/httpd.conf

Descomentar as linhas
Include etc/apache24/extra/httpd-ssl.conf
LoadModule ssl_module libexec/apache24/mod_ssl.so

cp nano /usr/local/etc/apache24/extra/httpd-ssl.conf nano
/usr/local/etc/apache24/extra/httpd-ssl.conf.BAK

Remova todo o conteúdo e adicione este

Mais prático

rm /usr/local/etc/apache24/extra/httpd-ssl.conf

nano /usr/local/etc/apache24/extra/httpd-ssl.conf

Listen 443
AddType application/x-x509-ca-cert .crt

```
AddType application/x-pkcs7-crl    .crl
<VirtualHost _default_:443>
  DocumentRoot "/usr/local/www/apache24/data"
  ServerName www.ribafs.org:443
  ServerAdmin ribafs@gmail.com
  ErrorLog "/var/log/httpd-error.log"
  TransferLog "/var/log/httpd-access.log"

  SSLEngine on

  SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:
+SSLv2:+EXP:+eNULL

  SSLCertificateFile "/usr/local/etc/apache24/ssl.crt/ribafs.crt"

  SSLCertificateKeyFile "/usr/local/etc/apache24/ssl.key/ribafs.key"

  <FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
  </FilesMatch>
  <Directory "/usr/local/www/apache24/cgi-bin">
    SSLOptions +StdEnvVars
  </Directory>

  BrowserMatch ".*MSIE.*" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0

  CustomLog "/var/log/httpd-ssl_request.log" "%t %h %{SSL_PROTOCOL}x %
{SSL_CIPHER}x \"%r\" %b"
</VirtualHost>
```

Dica: Cuidado com caracteres malucos criados com copiar e colar. Se precisar digite tudo.

service apache24 restart

Testar:
https://www.ribafs.org/info.php

https://www.rhyous.com/2009/11/06/installing-an-apache-ssl-on-freebsd-using-the-ports-tree/

Reforçando a Segurança do Apache

Configurações do Apache

Sobre o .htaccess
O arquivo .htaccess, que num Unix é um arquivo oculto (inicia com ponto), é um arquivo muito útil para a administração de segurança e vários outros recursos em sites que usam

o Apache como servidor web. Em especial quando não temos acesso direto às configurações do Apache.

Usado para configurar o Apache e também o PHP (somente se instalado como módulo do Apache).

Veja alguns dos seus vários e úteis usos.
Li certa vez: "Use e abuse do .htaccess, pois ele é seu amigo."

Listar Arquivos de Diretório
Se por exemplo você quer que o diretório onde você colocou o .htaccess liste os arquivos caso não haja um index.html da vida, você adiciona o seguinte no .htaccess:
Options +Indexes

Segurança no Apache/Nginx

ServerTokens Prod
ServerSignature Off
TraceEnable Off
Options all -Indexes
Header always unset X-Powered-By

E para tirar essa opção:
Options -Indexes

Permitir acesso somente para uma faixa de IPs:
<Files pagina_erro_403.php>
Order Deny,Allow
Deny from all
Allow from 192.168.
</Files>

Como personalizar páginas de erro:
ErrorDocument 403 /acesso_negado.php
ErrorDocument 404 /nao_encontrado.php
ErrorDocument 500 /erro_interno_servidor.php

401 - Authorization Required
400 - Bad request
403 - Forbidden
404 - Wrong page
500 - Internal Server Error

Ativar mod_rewrite
RewriteEngine On
RewriteCond %{SCRIPT_FILENAME} !-f
RewriteCond %{SCRIPT_FILENAME} !-d
RewriteRule ^(.*)$ index.php?pagina=$1

Bloqueia uma lista de IPs:
order allow, deny

deny from 210.140.98.160
deny from 69.197.132.70
deny from 74.14.13.236
allow from all

Deixa a Intranet acessar
Order allow,deny
allow from 192.168.0.
deny from all

Deixa todo mundo acessar, menos o IP 192.168.0.25
Order deny,allow
deny from 192.168.0.25
allow from all

Impedir todos de visitarem o site
deny from all
Obs.: no caso acima somente permite acesso pelo cpanel, ftp ou outro, nunca pela web.

Restringe o arquivo "secreto.html" somente para o IP 192.168.0.30
<Files secreto.html>
Order allow,Deny
Allow from 192.168.0.30
Deny from all
</Files>

Nega o acesso dos clientes ao .htaccess (bom colocar no httpd.conf)
Vem com a configuração padrão do Apache
<Files ~ "^\.ht">
Order allow,deny
Deny from all
</Files>

Redirecionar todos os visitantes que chegarem na pasta /antigo para o site
http://www.novosite.com/novo
Redirect /antigo http://www.site.com/novo

No caso, o arquivo http://www.site.com/antigo/teste.png será redirecionado para
http://www.site.com/novo/teste.png

Proteger Diretório com Login e Senha
Usar o comando htpasswd para criar a senha com a seguinte sintaxe:

htpasswd -c arquivodasenha login
htpasswd -c senha joao

O arquivo gerado conterá uma linha com login:senhacriptografada
Inserir no diretório um arquivo .htaccess com o conteúdo:

AuthName "Acesso Restrito"
AuthType Basic

```
AuthUserFile /backup/www/diretorio/senha
Require valid-user
```

/backup/www/diretorio/senha - é o caminho complato para o arquivo com a senha

Exige senha para acessar o diretório administrator

```
<Directory /administrator>
AuthName "Acesso Restrito à Usuários"
AuthType Basic
AuthUserFile /etc/httpd/auth/acesso
AuthGroupFile /etc/httpd/auth/grupos
require group admin
</Directory>
```

Bloquear Perl

Muitos scripts de ataque são criados em Perl, portanto para bloquear perl e outros bots para que não acessem seu site, adicione o código abaixo em um .htaccess (no raiz do domínio):

```
SetEnvIfNoCase User-Agent libwww-perl bad_bots
order deny,allow
deny from env=bad_bots
```

bogus handler para perl

Caso não esteja usando scripts em Perl em seu site adicione um "bogus handler" para estes scripts no .htaccess:

```
##Deny access to all CGI, Perl, Python and text files
<FilesMatch "\.(cgi|pl|py|txt)">
Deny from all
</FilesMatch>
## Se não está usando um arquivo robots.txt, então comente
# as 3 linhas abaixo para evitar o acesso somente ao arquivo robots.txt
<FilesMatch robots.txt>
Allow from all
</FilesMatch>
```

Outros recursos importantes para o .htaccess:

```
#Enable mod_rewrite and insert some sample rules:
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
#RewriteCond %{REQUEST_URI} ^(/component/option,com) [NC,OR] ##optional - see notes##
RewriteCond %{REQUEST_URI} (/|\.htm|\.php|\.html|/[^.]*)$ [NC]
RewriteRule ^(content/|component/) index.php
```

Protegendo o acesso direto ao .htaccess e ao configuration.php:

```
<Files .htaccess>
order allow,deny
deny from all
</Files>

<FilesMatch "configuration.php">
```

```
Order allow,deny
Deny from all
</FilesMatch>
```

E outros

```
<FilesMatch "\.(htaccess|htpasswd|ini|phps|log|sh|conf)$">
Order allow,deny
Deny from all
</FilesMatch>
```

Muito Importante
Adicione ao final do .htaccess:

```
# Block out any script trying to set a mosConfig value through the URL
RewriteCond %{QUERY_STRING} mosConfig_[a-zA-Z_]{1,21}(=|\%3D) [OR]
# Block out any script trying to base64_encode crap to send via URL
RewriteCond %{QUERY_STRING} base64_encode.*\(.*\) [OR]
# Block out any script that includes a <script> tag in URL
RewriteCond %{QUERY_STRING} (\<|%3C).*script.*(\>|%3E) [NC,OR]
# Block out any script trying to set a PHP GLOBALS variable via URL
RewriteCond %{QUERY_STRING} GLOBALS(=|\[|\%[0-9A-Z]{0,2}) [OR]
# Block out any script trying to modify a _REQUEST variable via URL
RewriteCond %{QUERY_STRING} _REQUEST(=|\[|\%[0-9A-Z]{0,2})
# Send all blocked request to homepage with 403 Forbidden error!
RewriteRule ^(.*)$ index.php [F,L]
# Dica de http://forum.codecall.net/security-tutorials/4867-joomla-hacking-script.html
```

Criando um sistema de detecção de intrusão com o .htaccess

URL encoding attacks such as SQL injection, white space, javascript, etc and redirects the URL to log.php. Log.php will then alert you via email.

Referência: http://www.hackosis.com/simple-htaccess-intrusion-detection-system/

```
Options +FollowSymLinks
RewriteEngine On
RewriteCond %{QUERY_STRING} (\"|%22).*(\>|%3E|<|%3C).* [NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteCond %{QUERY_STRING} (\<|%3C).*script.*(\>|%3E) [NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteCond %{QUERY_STRING} (javascript:).*(\;).* [NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteCond %{QUERY_STRING} (\;|\'|\"|\%22).*(union|select|insert|drop|update|md5|
benchmark|or|and|if).* [NC]
RewriteRule ^(.*)$ log.php [NC]
RewriteRule (,|;|<|>|'|`) /log.php [NC]
```

Criar o arquivo lop.php no raiz do site. Mudar o e-mail dmin@site.com Este endereço de e-mail está protegido contra spambots. Você deve habilitar o JavaScript para visualizá-lo para receber a notificação:

```
<?php
$r= $_SERVER['REQUEST_URI'];
```

```
$q= $_SERVER['QUERY_STRING'];
$i= $_SERVER['REMOTE_ADDR'];
$u= $_SERVER['HTTP_USER_AGENT'];
$mess = $r . ' | ' . $q . ' | ' . $i . ' | ' .$u;
mail(" admin@site.com este endereço de e-mail está protegido contra spambots. Você deve habilitar o JavaScript para visualizá-lo. ","bad request",$mess,"from: bot@site.com este endereço de e-mail está protegido contra spambots. Você deve habilitar o JavaScript para visualizá-lo. ");
echo "Hot Damn!";
?>
```

Reforçar a Segurança do PHP

Reforçar a segurança do PHP

Uma boa forma de melhorar a segurança do php é instalando o phpsecinfo:
https://github.com/funkatron/phpsecinfo
http://phpsec.org/projects/phpsecinfo/

E corrigir os erros apontados com as respectivas recomendações.

Algumas sugestões para reforçar a segurança do PHP:

edite o php.ini e faça as alterações:
nano /etc/php.ini

ALERTA – ao efetuar as alterações abaixo faça uma a uma, sempre reiniciando o apache e abrindo o site e efetuando um refresh para testar. Caso tenha problema desfaça ou ajuste o parâmetro com problema.

disable_functions = exec,system,shell_exec,passthru,
html_errors = Off
mail.add_x_header = Off
session.name = NEWSESSID

Na linha com disable_functions já existem várias funções por padrão que são desabilitadas. Não as remova, apenas adicione as recomendações acima ao início, separadas por vírgula.

Com a ajuda do PHPsecinfo também ajustei estes abaixo:

allow_url_fopen = Off
upload_tmp_dir = /usr/share/nginx/html/phpup

Criei o diretório /usr/share/nginx/html/phpup

Estes dois últimos parâmetros devem ser adotados com cuidado, de acordo com a sua necessidade. Abaixo são os valores default na versão 7 do php:
post_max_size = 8M
upload_max_filesize = 2M

service nginx restart
Depois dos ajustes acima alguma coisa pode não funcionar. Então efetue os ajustes devidos, sem exagerar.

Proteger arquivos de configuração do apache, php e mysql contra escrita:

/etc/php//php.ini
/etc/nginx/conf.d/default.conf e demais
/etc/mysql/my.cnf

Configurações do PHP
php.ini e ini_set()

display_errors
Estes só devem estar ativos quando estamos em ambiente de testes, criando ou programando, antes de enviar para o servidor devemos desativar. Se alguma extensão ou aplicativo precisa dessas extensões ativas o prudente é não usar essas extensões. Aliás, para quando estamos instalando, testando e programando, devemos exibir ao máximo os erros. Idealmente o php.ini deve estar com:

error_reporting = E_ALL & ~E_DEPRECATED

Quando usar php.ini, ini_set() ou .htaccess
Isso vai depender de como o servidor está configurado.
- Algumas configurações do PHP podem ser alteradas usando a função ini_set(), sendo que as alterações são válidas somente enquanto o script estiver em execução.
- Outras configurações devem ser adicionadas num script php.ini ou no .htaccess.
- Outras em qualquer uma das formas acima.
- E algumas configurações não podem ser alteradas, como é o caso de memory_limit e execution_time. Caso alteremos essas podemos prejudicar a performance do site.
Quando e onde usar cada um vai depender de como o PHP está instalado no servidor, que pode ser como módulo CGI ou como um módulo do Apache.

PHP como Módulo do Apache ou como CGI
Para saber como foi instalado o PHP no seu servidor poderá consultar o help desk. Alguns servidores armazenam as respostas do suporte numa área chamada de Knowledge Base até com busca.

A instalação como módulo CGI é mais segura, portanto preferível. Alguns servidores oferecem as duas modalidades e geralmente para servidores compartilhados é oferecido como módulo CGI, mas para ter certeza pergunte ou faça testes para identificar. Veja detalhes abaixo.

Como CGI - mais segura, portanto mais restritiva para alterar as configurações do PHP. Não podemos criar um php.ini no raiz e ele valer recursivamente para todos os sub-diretórios. Temos que criar um php.ini para cada diretório.
Também não podemos adicionar configurações do PHP nos scripts .htaccess. Se o fizermos receberemos um erro e o script não funcionará. Temos que remover o que adicionamos para o site voltar. Podemos usar a função ini_set().

Como módulo do Apache - Aqui podemos criar um php.ini no raiz do domínio e ele valerá para todos os sites, ou seja, tem efeito global. Também podemos usar as configurações do PHP em scripts .htaccess e podemos usar a função ini_set().

Como Configurar o PHP
ini_set() - Esta função é muito útil para efetuar configurações em servidores onde o PHP foi instalado como CGI, especialmente para sites com Joomla. No caso entramos com as linhas da ini_set() no configuration.php, que é visto por todos os scripts.

Exemplo usando ini_set()

```
ini_set('extension', 'sourceguardian.so');
ini_set('session.save_path', '/home/joao/public_html/tmp');
ini_set('cgi.force_redirect', 1);
ini_set('allow_url_fopen', 0);
ini_set('display_errors', 0);
ini_set('allow_url_include', 0);
ini_set('expose_php', 0);
ini_set('magic_quotes_gpc', 0);
ini_set('post_max_size', 262144);
ini_set('upload_max_filesize', 262144);
ini_set('upload_tmp_dir','/home/joao/public_html/tmp');
$disfunctions = 'proc_open, popen, disk_free_space, set_time_limit, leak, tempfile, exec,
system, shell_exec, passthru, curl_exec, curl_multi_exec, parse_ini_file, show_source,
apache_get_modules, apache_get_version, apache_getenv, apache_note,
apache_setenv, disk_free_space, diskfreespace, dl, highlight_file, ini_alter, ini_restore,
openlog, proc_nice, symlink, phpinfo';
ini_set('disable_functions', $disfunctions);

// Verifique se pode usar estes abaixo em seu servidor
ini_set('zend_extension', '/usr/local/php52/lib/php/extensions/ioncube.so');
ini_set('zend_extension_manager.optimizer=', '/usr/local/Zend/lib/Optimizer-3.3.3');
ini_set('zend_extension_manager.optimizer_ts', '/usr/local/Zend/lib/Optimizer_TS-3.3.3');
ini_set('zend_optimizer.version', '3.3.3');
ini_set('zend_extension', '/usr/local/Zend/lib/ZendExtensionManager.so');
ini_set('zend_extension_ts', '/usr/local/Zend/lib/ZendExtensionManager_TS.so');
```

php.ini - Quando criamos um php.ini e adcionamos a um certo diretório, as diretivas dele sobrescreverão as existentes no script php.ini do servidor, mudando o valor das diretivas, mas perdendo alguns recursos importantes, como é o caso do ionCube. Veja exemplo abaixo para contornar isso.

Exemplo de php.ini para reforçar a segurança

```
extension=sourceguardian.so
session.save_path = "/home/ribafs03/public_html/tmp"
cgi.force_redirect = 1
allow_url_fopen= 0
display_errors = 0
expose_php = 0
magic_quotes_gpc = 0
memory_limit = 8388608
#open_basedir = 1
post_max_size = 262144
```

upload_max_filesize = 262144
upload_tmp_dir = "/home/ribafs03/public_html/tmp"
disable_functions = proc_open, popen, disk_free_space, set_time_limit, leak, tempfile, exec, system, shell_exec, passthru, curl_exec, curl_multi_exec, parse_ini_file, show_source, apache_get_modules,apache_get_version,apache_getenv,apache_note, apache_setenv, disk_free_space, diskfreespace, dl, highlight_file, ini_alter, ini_restore, openlog, proc_nice, symlink, phpinfo

// Checar se seu servidor suporta os abaixo e as versões
zend_extension=/usr/local/php56/lib/php/extensions/ioncube.so
zend_extension_manager.optimizer=/usr/local/Zend/lib/Optimizer-3.3.3
zend_extension_manager.optimizer_ts=/usr/local/Zend/lib/Optimizer_TS-3.3.3
zend_optimizer.version=3.3.3
zend_extension=/usr/local/Zend/lib/ZendExtensionManager.so
zend_extension_ts=/usr/local/Zend/lib/ZendExtensionManager_TS.so

Lista de todas as diretivas só php.ini
http://www.php.net/manual/pt_BR/ini.list.php
Descrição das diretivas do principais do arquivo php.ini
http://www.php.net/manual/pt_BR/ini.core.php

Reforçando a Segurança do MySQL

Melhorando a Segurança do MySQL/MariaDB

Uma forma de melhorar a segurança do mysql é criar usuários restritos, que somente tenham poder de agir num banco específico.

O exemplo abaixo é usado para criar um usuário a ser usado em site com Joomla:

mysql -u root -p
create database portal;

GRANT ALL PRIVILEGES ON portal.* TO portal@localhost IDENTIFIED BY 'senha' WITH GRANT OPTION;
\q

Alerta
Geralmente se cria usuário deste forma, concedendo todos os privilégios ao usuário, mas para melhorar a segurança veja o exemplo abaixo onde somente concedemos os privilégios que o usuário ncessitará.

Importar Script:
mysql -u root -p portal < portal.sql

Exportar banco para script:
mysqldump -u root -p portal > portal.sql

Também importante é executar

mysql_secure_installation

Criando um banco e um usuário para ele

mysql -u root -p

CREATE DATABASE familia CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE USER 'us_familia'@'localhost' IDENTIFIED BY 'senhaforte';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER,
CREATE TEMPORARY TABLES, LOCK TABLES ON familia.* TO 'us_familia'@'localhost'
IDENTIFIED BY 'yourpassword';
FLUSH PRIVILEGES;
\q

Outros privilégios

ALL PRIVILEGES- como vimos anteriormente, isso daria a um usuário do MySQL todo o acesso a uma determinada base de dados (ou se nenhuma base de dados for selecionada, todo o sistema)
CREATE- permite criar novas tabelas ou bases de dados
DROP- permite deletar tableas ou bases de dados
DELETE- permite deletar linhas das tabelas
INSERT- permite inserir linhas nas tabelas
SELECT- permite utilizar o comando Select para ler bases de dados
UPDATE- permite atualizar linhas das tabelas
GRANT OPTION- permite conceder ou revogar privilégios de outros usuários

mysql -u root -p
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER,
CREATE TEMPORARY TABLES, LOCK TABLES ON joomla.* TO
'yourusername'@'localhost' IDENTIFIED BY 'yourpassword';

Melhorando a segurança de sites com Joomla

Atualização - Mantenha o Joomla e as extensões do seu site atualizados.

Recomendação importante: sempre faça um backup full do site (todos os arquivos e o banco), antes de atualizar, pois pode ser que você tenha alterado alguma extensão ou o próprio Joomla e a atualização apague isso. Sugestões: com_simplebackup e AkeebaBackup.

Não atualize para novas versões (exemplos: da 2.5 para a 3, da 3.x para a 4) de imediato, tenha prudência

- Uma ótima extensão para colaborar com essa tarefa é o componente Admin Tools, do mesmo desenvolvedor do Akeda Backup: http://www.akeebabackup.com/download.html
- Sempre instale imediatamente que faça o download, mas antes de atualizar realize um backup full para se prevenir, pois algumas atualizações podem alterar o template default

(que você pode estar usando) ou outra extensão que tenha personalizado, portanto backup full antes.

Aproveite para assinar também o RSS das novidades sobre segurança:
http://feeds.joomla.org/JoomlaSecurityNews

Criação do banco e do usuário para Joomla

mysql -u root -p

CREATE DATABASE db_cakephp CHARACTER SET utf8 COLLATE utf8_general_ci;

GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES ON db_cakephp.* TO 'us_cakephp'@'localhost' IDENTIFIED BY 'senhaforte';

FLUSH PRIVILEGES;

Extensões de Terceiros
Evite ao máximo utilizar extensões de terceiros. Sempre que o Joomla tiver uma extensão, prefira a do Joomla, como por exemplo, para URLs amigáveis prefira usar o recurso do Joomla. Somente instale extensões de terceiros se extremamente necessário. E se instalar sempre vá ao site do autor e baixe a última versão e também mantenha-a atualizada. Também tenha o cuidado de instalar antes em um computador de testes e somente após alguns testes instale no site em produção.

Muita prudência ao instalar extensões de terceiros:
- Evite usar extensões de terceiros, especialmente quando o Joomla já trouxer uma nativa similar.
- Caso decida instalar visite antes a Lista de Vulnerabilidade de Extensões.
- Extensões não usadas devem ser desinstaladas.
- Instale em máquina de teste antes de colocar em produção. Instale várias vezes, teste, execute e se for um template teste em vários navegadores.
- Baixe somente do site dos criadores
- Mantenha sempre atualizada
- Caso descubra que uma extensão é insegura não somente desabilite mas desinstale e remova manualmente tudo que restar

Evite Alteração do Código do Core
Evite hackear (alterar) o código do core do Joomla tornando difícil a manutenção e atualização e inseguro o mesmo.

Criação de Artigos
Ao criar artigos ou permitir que estranhos criem artigos é importante filtrar HTML ou configurar o usuário para não usar editor HTML, sob pena de ter o site invadido.

Download do Joomla
Faça sempre o download do site oficial.

Valores Default

Valores default são perigosos, pois são de conhecimento de todos.

Esconda o diretório administrator de curiosos usando uma extensão como oAdminExile.

Mover o configuration.php para fora da área do Apache

Mover o configuration.php para fora da área do Apache e mudar as permissões para 400. Caso queira alterar mude para 600.
Se seu site estiver em public_html/portal
Então copie o configuration.php para o diretório abaixo de public_html e o renomeie para portal.cfg, além de mudar as permissões dele para 400.
Somente mude para 600 se precisar alterar. Então edite o original deixando somente a linha abaixo:
require_once(dirname(__FILE__) . '/../../portal.cfg');

URLs Amigáveis
Ativar as URLs amigáveis para evitar ataques pela URL e também maior visibilidade no Google além de tornar mais amigável para os visitantes.

robots.txt
Edição de robots: libere a pasta images

Páginas de Erro
Crie páginas de erro mais bonitas e com informações úteis ao visitante, como um e-mail para feedback.
Joomla não é Perfeito
O Joomla é o software mais bem feito que já vi, mas cuidado para não chegar a pensar que o Joomla é perfeito e não precisa de ajustes na segurança.

Logs e Tmp
Diretório de Logs - Altere no Admin: Configuração Global - Sistema - Caminho para o diretório do log

Diretório Tmp - Altere também: Configuração Global - Servidor - Caminho para o diretório temporário.

Desabilitar Extensões não Usadas
Plugin XML-RPC caso não precise dele desabilite-o. Vem desabilitado por default.

O site está em
/var/www/html/portal

- Copiar configuration.php para o /var/www com o nome cfg.php
- Remover todo o conteúdo do /portal/configuration.php e deixar apenas estas duas linhas:

<?php
require_once(dirname(__FILE__) . '/../../cfg.php');

Obs.: lembre de fazer o backup do arquivo cfg.php, que agora está fora do html.

Algumas Extensões de Terceiros que Colaboram com a Segurança (lista não atualizada)

AdminTools - Componente que detecta novas atualizações do Joomla e do próprio componente, alertando quando o acessamos. Além disso ele pode instalar a nova versão com apenas 3 cliques. Além disso ele também corrige as permissões do site para 755 (diretórios) e 644 (arquivos), adiciona uma segunda camada de segurança (novo login) ao administrator, além de outros recursos.
É um verdadeiro canivete suiço para sites em Joomla.
Atualmente recebemos um e-mail tão logo sai uma nova versão do Joomla. Ao ser avisado devemos ir ao administrator e o atualizar.
Recomendação importante: sempre faça um backup full do site (todos os arquivos e o banco), antes de atualizar, pois pode ser que você tenha alterado alguma extensão ou o próprio Joomla e a atualização apague isso e eventualmente a atualização pode impedir o acesso ao site.

Plugin jHackGuard - plugin criado pelo SiteGroun sites para protejer sites em Joomla de ataques de hackers. Basta instalar e ativar para ganhar proteção contrar ataques tipo SQL Injections, Remote URL/File Inclusions, Remote Code Executions e XSS.
http://extensions.joomla.org/extensions/access-a-security/site-security/site-protection/13233

Veja este artigo do Júlio Coutinho sobre o JhackGuard:
JHackGuard plugin de segurança para o Joomla!

jHackGuard é um plugin desenvolvido pelo SiteGround e ajuda a proteger contra ataques de crackers. Basta adicionar ao seu Joomla e ele estará mais seguro contra sqlinjections, codeinjections e ataques baseados em xss!
Este plugin tem sido utilizado com sucesso pelos clientes SiteGround durante os últimos anos e o Siteground resolveu tornar pública a sua versão mais recente, de modo que você pode facilmente proteger o seu site Joomla. Tudo que você precisa fazer é instalar jHackGuard e habilitá-lo - nenhuma configuração adicional necessária!
O Plugin de segurança do SiteGround assegura sites Joomla, protegendo-os contra a diferentes técnicas de hacking. Ele filtra os dados de entrada dos usuários e implementa as configurações de segurança adicionais PHP.
O Plugin de segurança do SiteGround contém opções de filtragem de segurança avançadas. Ele vem com um conjunto de regras pré-definidas, que funciona na maioria dos casos. Caso deseje, você pode alterar os padrões. Felizmente, ele tem um log e você pode depurar qualquer comportamento inesperado.
O Plugin de segurança do SiteGround pode ser configurado através da área de administrador do Joomla. O plugin está desativado para os administradores autenticados para que os filtros não os impeçam de fazer tarefas administrativas.

Aconselho testar o plugin localmente e caso haja alguma instabilidade em seu website, faça o seguinte:
1) Acesse o banco de dados pelo phpmyadmin
2) Localize a tabela #__plugins
3) Localize o plugin JHackGuard
4) Clique no lápis (editar)
5) Altere o valor do status de 1 para 0 para desabilitar o plugin

Instalação jHackGuard
A instalação do plugin é padrão como qualquer outra extensão do Joomla.
Você baixa o plugin para sua máquina e no backend do Joomla navega pelo menu superior Extensões -> Instalar / Desinstalar.
Clique no botão Procurar e localize o pacote de extensão no seu disco rígido. Em seguida, clique no botão "Upload File & Install". O plugin será instalado e acrescentado ao Joomla.

Por último, siga por Extensões -> Gerenciamento de Plugin. Nela localize o plugin jHackGuard e clique no ícone "habilitar". Isso irá ativar o plugin.
As regras padrão de jHackGuard foram programadas pelos especialistas do Siteground, com base em sua experiência na fixação de um grande número de vulnerabilidades de diferentes sites Joomla. Recomendamos o uso das normas padrão para o melhor desempenho do plugin.

No entanto, se você quiser fazer alterações específicas para suas configurações, você pode fazer isso a partir da página Gerenciamento de Plugin em sua área administrativa do Joomla. Uma vez lá, clique no secuity - rótulo Plugin jHackGuard para entrar em sua página de configurações. Os parâmetros configuráveis para o plugin são separados em vários grupos:
* Opções de Login
* Arquivo de log - Aqui você pode digitar o nome do arquivo onde os registros sobre as atividades plugin serão mantidos. O nome do arquivo padrão é log.php-jHackGuard. Ela é armazenada sob a pasta de logs.
* Enable Login - Você pode decidir se as atividades do plugin serão registradas
* Fluxos de Dados
* Filtro $ _POST - Filtros variáveis provenientes do método POST HTTP.
* Filtro $ _GET - Filtros de variáveis passadas para o script através de parâmetros de URL.
* Filtro $ _COOKIE - Filtros de variáveis provenientes de HTTP Cookies.

* Parâmetros de filtragem
* Filtro de eval () - Filtra o resultado da avaliação de uma string como código PHP.
* Filtro base64_decode - Filters o resultado da decodificação de dados codificados em base64.
* Filtro de comandos SQL - Filtros a execução de comandos SQL. Esta solução evita os ataques de injeção SQL.

* Parâmetros avançados
* Allow_url_fopen - Desativa a opção de recuperar arquivos de FTP remoto ou servidor web. Esta solução protege seu site contra injeção de código.
* Allow_url_include - Desativa a opção de incluir URLs de pedidos PHP. Desta forma seu site estará protegido contra ataques remotos URL Inclusão.
(*) Plugin sugerido por LinkDF 2010, membro da comunidade Joomla! Brasília no Orkut.
Fonte: http://www.siteground.com

Artigo traduzido e adaptado por Júlio Coutinho no site:
http://www.joomlabrasilia.org/tutoriais-de-joomla/seguranca-e-joomla/624.html

Este plugin adiciona um link na parte inferior do template e tem um pequeno problema para quem usa o PHP mostrando Notices.
Comente a linha 80 caso queira ocultar o link.

A linha 370 deve ficar assim:
$chars = 'PCRE_UNICODE_PROPERTIES' ? '\pL' : 'a-zA-Z';
Adicionando as aspas simples na contante.

Componente com_encrypt – Adiciona criptografia para os campos dos forms, o que evita que usuários maliciosos monitorem e capturem senham em texto claro.

Vários outros para extensões de terceiros:
http://www.ratmil.com/downloads/category/4-encryption-configuration-plugins.html

Plugin Marcos Interceptor SQL Injection – Um plugin para prevenir SQL injection, prevenindo ataques deste tipo, contendo:
•.Filters requests in POST, GET, REQUEST. and blocks SQL injection / LFI attempts
•.Notifies you by e-mail when a alert is generated
•.Protect also from unKnown 3rd Party extensions vulnerability.
•.White list for safe components (at your risk ;))
Enable mail report and prepare yourself to be scared!
Anyway remember that security it is a 'forma mentis', not a plugin!
http://www.mmleoni.net/sql-iniection-lfi-protection-plugin-for-joomla

OSOL Captcha - Use um bom plugin com Captcha para todos os forms do site.
Este adiciona captcha em qualquer form do site.

Para adicionar um captcha em form que não adicionou automaticamente, adicione no código do form:

```
<?php
global $mainframe;
//set the argument below to true if you need to show vertically( 3 cells one below the other)
$mainframe->triggerEvent('onShowOSOLCaptcha', array(false));
?>
```

Mude false para true se quiser os campos na vertical.
http://www.outsource-online.net/osol-captcha-for-joomla.html

Ferramentas
Para Joomla, ferramentas são softwares que não têm um instalador como as demais instalações mas que de alguma forma trabalham com o Joomla, sendo executados pela linha de comando ou mesmo pela web, como as abaixo.

Joomla Scan
Ferramenta não instalável, que precisa ser executada na linha de comando.
Yet Another Joomla Vulnerability Scanner that can detects file inclusion, sql injection, command execution vulnerabilities of a target Joomla! web site.

Download
http://sourceforge.net/projects/joomscan/
http://www.owasp.org/index.php/OWASP_Joomla_Vulnerability_Scanner_Usage
http://www.owasp.org/index.php/Category:OWASP_Joomla_Vulnerability_Scanner_Project

Descompacte para o public_html ou /var/www/html
Acesse o terminal e vá para a pasta joomscal-latest

chmod +x joomscan.pl
./joomscan.pl -pv -u victim.com -x localhost
Irá mostrar vulnerabilidades encontradas no Joomla e as respectivas versões

phpSecInfo – este é um aplicativo em PHP que deve ser instalado no servidor (não usa bancos de dados) e mostra as vulnerabilidades do PHP e sugestões para corrigir.
http://phpsec.org/projects/phpsecinfo/

Resetar senha de usuário da seção administrator do Joomla
Execute o phpmyadmin ou o adminer
Abra o banco do portal
Selecione a tabela jos_users
Edite o registro do usuário que deseja resetar a senha, geralmente o super-administrador
Apague todo o conteúdo do campo password e digite a senha deseja.
Em Funções selecione MD5 para o campo password
Clique no botão executar
Agora pode acessar o administrator

Checklist de Segurança para Joomla

Listagem rápida para checar em sites antes de serem colocados em produção:
- Efetue um backup completo de todos os arquivos e do banco e restaure localmente
- Ativar URLs amigáveis e mod_rewrite
- Mover configuration.php para fora do public_html, usando:
require_once(dirname(__FILE__) . '/../../portal.cfg');
- Bloquear cadastro de usuários pelo site caso não tenha necessidade: Configuração Global - Sistema - Permitir Cadastro de Usuários - Não
- Alterar metatags em Configuração Global - Configurações de Meta Dados (Trocar Joomla por outra palavra)
- Adicionar a tag <head> do template (para ocultar na origem do código HTML):
<?php $this->setGenerator('Ribafs - Desenvolvimento Web'); ?>
- Instalar algumas extensões:
- AdminTools
- Plugin osolcapcha
- com_encrypt
- jHackGuard

Usar ferramentas:
joomscan
Ativar o cache
Otimizar as tabelas do banco no phpmyadmin

Sanear permissões de arquivos:
Alterar todos os arquivos recursivamente para 644 e todas as pastas para 755 com. Veja em Permissões como fazer isso.

Depois criar algumas exceções...
configuration.php – 400
index.php do site – 400
index.php do template padrão – 400

Permissões de pastas:
includes e libraries – 500

Remover templates não usados e outras extensões também.

Adicionar ao .htaccess:

```
# Block out any script trying to set a mosConfig value through the URL
RewriteCond %{QUERY_STRING} mosConfig_[a-zA-Z_]{1,21}(=|\%3D) [OR]
# Block out any script trying to base64_encode crap to send via URL
RewriteCond %{QUERY_STRING} base64_encode.*\(.*\) [OR]
# Block out any script that includes a <script> tag in URL
RewriteCond %{QUERY_STRING} (\<|%3C).*script.*(\>|%3E) [NC,OR]
# Block out any script trying to set a PHP GLOBALS variable via URL
RewriteCond %{QUERY_STRING} GLOBALS(=|\[|\%[0-9A-Z]{0,2}) [OR]
# Block out any script trying to modify a _REQUEST variable via URL
RewriteCond %{QUERY_STRING} _REQUEST(=|\[|\%[0-9A-Z]{0,2})
# Send all blocked request to homepage with 403 Forbidden error!
RewriteRule ^(.*)$ index.php [F,L]
```

Adicionar ao configuration.php:

```
ini_set('extension', 'sourceguardian.so');
ini_set('session.save_path', '/home/ribafs03/public_html/tmp');
ini_set('cgi.force_redirect', 1);
ini_set('allow_url_fopen', 0);
ini_set('display_errors', 0);
ini_set('allow_url_include', 0);
ini_set('expose_php', 0);
ini_set('magic_quotes_gpc', 0);
ini_set('post_max_size', '262144');
ini_set('upload_max_filesize', '262144');
ini_set('upload_tmp_dir','/home/joao/public_html/tmp');
$disfunctions = 'proc_open, popen, disk_free_space, set_time_limit, leak, tempfile, exec,
system, shell_exec, passthru, curl_exec, curl_multi_exec, parse_ini_file, show_source,
apache_get_modules, apache_get_version, apache_getenv, apache_note,
apache_setenv, disk_free_space, diskfreespace, dl, highlight_file, ini_alter, ini_restore,
openlog, proc_nice, symlink, phpinfo';
ini_set('disable_functions', $disfunctions);
// Verificar se seu servidor duporta e ajustar as versões
ini_set('zend_extension', '/usr/local/php56/lib/php/extensions/ioncube.so');
ini_set('zend_extension_manager.optimizer=', '/usr/local/Zend/lib/Optimizer-3.3.3');
ini_set('zend_extension_manager.optimizer_ts', '/usr/local/Zend/lib/Optimizer_TS-3.3.3');
ini_set('zend_optimizer.version', '3.3.3');
ini_set('zend_extension', '/usr/local/Zend/lib/ZendExtensionManager.so');
ini_set('zend_extension_ts', '/usr/local/Zend/lib/ZendExtensionManager_TS.so');
```

E se o site foi invadido?

Devemos sempre prevenir. Mas e se acontecer de o site ser invadido, o que fazer?
- Mantenha o site offline para evitar outros ataques
- Baixe a última versão do Joomla
- Notifique o suporte do servidor e trabalhe com ele para fazer a limpeza do site e para ter certeza de que não ficou nenhum back door no site.
- Visite novamente o site das vulnerabilidades e veja se você ainda tem alguma extensão vulnerável
- Mude todas as senhas e se possível logins: cPanel, mysql, FTP, joomla Super Admin, etc.
- Substitua todos os templates e arquivos por cópias limpas
- Verifique atentamente os arquivos de log
- Verifique o cron se tem alguma entrada estranha
- Preferivelmente remova todo o conteúdo adicionado e todos os bancos e todos os e-mails para criar tudo novo
- Restaure com backups bem antes do ataque
- Confira as permissões de todos os arquivos. Nunca use 777, somente 644 para arquivos e 755 para diretórios
- Caso tenha acesso via SSH execute os comandos a seguir para sanear os arquivos:
find /home/joao03/public_html/site -type f -exec chmod 644 {} \;
find /home/joao03/public_html/site -type d -exec chmod 755 {} \;

Quando temos um site invadido, existe uma grande chance do invasor ter deixado backdoors para poder voltar depois. A procura pelos backdoors é algo demorado e trabalhoso, inclusive sem garantia, por isso é mais prudente remover tudo e instalar do zero, sem contar a hipótese de mudar de servidor, caso suspeite da fragilidade da sua hospedagem.

Para checar os arquivos recentemente alterados no sistema
find \public_html -ctime -1

Para proteger diretórios que precisam de premissão 777 ou por default
Crie um .htaccess no diretório images com:
secure directory by disabling script execution
AddHandler cgi-script .php .pl .py .jsp .asp .htm .shtml .sh .cgi
Options -ExecCGI

Varrer Servidor
Faça uma varredura em todos os sites/máquinas a procura de malware, virus, trojans, spyware, etc.
* Opções disponíveis:
- ENOD32 from eSet
- Spybot Search and Destroy
- Malwarebytes
- Microsoft Malicious Software Removal Tool
- Linux AntiVirus boot cd
* Considere o Ultimate Boot CD para Windows

O que é uma extensão vulnerável?
- É uma extensão que contém ou contribui para uma vulnerabilidade de segurança.
- Quanto mais complexo for seu código maior a chance de ser vulnerável.

- As que lêem e escrevem arquivos
- Se não valida entradas de usuários
- Usam path explícito
- Permite acesso direto pela URL
- Permite inclusão de arquivos remotos
- Permite SQL injections
- Permite XSS
- Permite muito acesso a banco para usuários sem privilégios

Extensões vulneráveis são, não necessariamente extensões com código malfeito. Projetos ativos geralmente lançam novas versões de suas extensões com as alterações. Por estas razões é importante:
- Conhecer o número da versão de todas as extensões instaladas
- Use somente a última versão estável das extensões
- Desinstale e remova completamente todos os arquivos de extensões inseguras
Caso a última versão estável tenha sido lançada há um ano ou mais considere o projeto abandonado. Não instale extensões antigas.

Fórum de Ajuda do Google para Webmasters
•.Aprenda com outros usuários
•.Otimização do mecanismo de pesquisa
Aprimore o desempenho do seu site em pesquisas [PDF]
•.Ferramentas de terceiros para Sitemaps
Ferramentas para a criação de Sitemaps

Desktop

Melhorar a segurança no Desktop

Precisamos ter nosso ambiente de trabalho, nosso computador desktop onde criamos o site ou preparamos a instalação do servidor, precisa estar isento de ameaças, para que ao enviar o conteúdo ou o nosso site para o servidor não estejamos colaborando para aumentar as ameaças do mesmo. Por isso procede a sugestão de usar um sistema operacional mais seguro como Linux. Mesmo usando Linux devemos usar um bom firewall para filtrar ameaças. Caso utilizemos Windows devemos nos prevenir usando um navegador menos inseguro e com um bom e atualizado anti-virus, firewall e diversos outros softwares que ajudem a manter o ambiente limpo de ameaças.

Fique atento para a atualização de todos os softwares importantes que utiliza, como antivirus, firewall, IDEs, etc. Não esquecer de atualizar o Sistema Operacional.
Use sempre senhas fortes para tudo no servidor e inclusive em seu desktop. Será perca de tempo investir em muitos cuidados com a segurança, muito tempo de trabalho, muita pesquisa e estudo, se usar senhas fracas e fáceis de serem quebradas. Senhas fortes devem ter no mínimo letras e números. Para reforçar use também símbolos. Uma recomendação importante é que nunca mantenha sua senha do servidor em arquivo texto.

Ajuda muito usar com frequência programas/sites para varredura/scan dos sites que estamos trabalhando. Veja na seção de programação algumas sugestões.

Mesmo que você esteja usando Linux, instale um antivirus como o clamav para manter sua máquina limpa de arquivos de outros sistemas operacionais frágeis e para varrer os arquivos do site quando baixar e antes de enviar. Não esquecer de varrer pendrives que vieram do Windows.

Melhorar a segurança no desktop é importane para maior segurança do servidor. Hábitos saudáveis como usar um sistema operacional seguro e atualizado, como usando o firewall ativo e fechando tudo que pode.

Assim como também instalando boas ferramentas de monitoramento do servidor.

Instalar no micro desktop o W3AF

apt-get install w3af

Traz uma interface para a console e uma gráfica/web

Testando vulnerabilidades web com Nikto

O Nikto é web server scanner escrito em perl usado para detectar vulnerabilidades em servidores web. Ele é muito simples de ser usado e atualizado gerando relatórios em txt, html e csv.

Requer repositório multiverse no /etc/apt/sources.list

apt-get install nikto

Atualizando os plugins:

nikto -update

Usando o Nikto

nikto -h HOST -p PORT
nikto -h HOST -p PORT -ssl
nikto -h ribafs.org
nikto -C all -host 200.128.X.X -o vitima.txt (mude X.X pelos números desejados)

- C all - Força a checagem de todos os diretórios em busca de cgi
- host - Ip da vitima
-o - Gera um arquivo de relatório

Varrendo uma porta de um host:
nikto -h google.com -p 443

Help
nikto -H | less

Esta ferramenta tanto ajuda a defender o seu site quanto ajuda para quem quer perceber vulnerabilidades em outros sites ou atacar.

Documentação oficial:
http://cirt.net/nikto2-docs/

Exemplos de uso:
http://cirt.net/nikto2-docs/usage.html

Clamav

O clamav precisa estar instalado no desktop caso se use arquivos do windows no desktop.

sudo su
apt update

apt-cache search clamav

apt install clamav clamav-daemon
freshclam

Checando
clamscan -r /home/ribafs
clamscan -r /

Todo o computador
clamscan -r --bell -i /

Criar lista de arquivos infectados
clamscan -r /home/ribafs/ | grep FOUND >> report.txt

Versão
clamdscan -V

Adicionando ao cron

crontab -e

00 00 * * * clamscan -r /home

Instalar gui
apt-get install ClamTK

https://www.unixmen.com/installing-scanning-clamav-ubuntu-14-04-linux/

Usando o nmap

O Nmap é um portscan de uso geral, que pode ser usado, sempre que você precisar verificar rapidamente as portas abertas em determinado host, seja na sua rede local, seja na Internet.

Instalar no desktop para monitorar o servidor.

sudo apt install nmap

O uso mais simples é escanear diretamente uma máquina da rede, como em:
nmap 192.168.0.3

nmap 177.130.208.59

Starting Nmap 7.01 (https://nmap.org) at 2018-03-26 19:42 -03
Nmap scan report for 177.130.208.59
Host is up (0.052s latency).
Not shown: 994 closed ports
PORT STATE SERVICE
22/tcp filtered ssh
23/tcp filtered telnet
80/tcp open http
139/tcp filtered netbios-ssn
8082/tcp filtered blackice-alerts
12345/tcp filtered netbus

Nmap done: 1 IP address (1 host up) scanned in 1.85 seconds

Veja que se refere a 994 portas fechadas. Ele varreu as 1000 primeiras portas.

Mostra várias portas abertas.
O simples fato de uma determinada porta estar aberta, não significa que a máquina está vulnerável, mas apenas que existem serviços ativos e as portas não estão sendo bloqueadas por nenhum firewall.

Mostrar o sistema operacional
sudo nmap -O IP_dominio

nmap -v -A ribafs.org

nmap -v -sn ribafs.org

Scan completo em todas as portas:
nmap -sS -p 0-65535 192.168.0.4

sudo nmap -sS -p 0-65535 - em servidor com freebsd e ipfw não conseguiu. Sugeriu usar -Pn mas demorou demais
sudo nmap -sS -Pn -p 0-65535 IP

Scannear uma porta específica

nmap -sV -p 22543 192.168.0.4
nmap -p 80 192.168.2.2
nmap -p 80,443 192.168.1.1
nmap -p 80-200 192.168.1.1

Forçar scan de serviços escondidos em portas altas
nmap -sS -P0 -p 0-65535 192.168.0.4

Analisar servidor protegido por firewall
nmap -PN 192.168.1.1

Analisar que programas e versão correm nas portas abertas:
nmap -sv 192.168.2.2

Procurar falhas na firewall
Uma analise nula para fazer a firewall gerar uma resposta##
nmap -sN 192.168.2.2

Verificação de firewall
nmap -sF 192.168.2.2

Faz os sets FIN, PSH, e URG, serem analisados.
nmap -sX 192.168.2.2

A opção -D faz com que o alvo pense que está a ser analisado por mais maquinas.
#O IDS fará com que se reporte entre 5 a 10 portas a cada IP mas nunca sabe quais são os verdadeiros e os falsos.
nmap -n -Ddecoy-ip1,decoy-ip2,your-own-ip,decoy-ip3,decoy-ip4 remote-host-ip
nmap -n -D192.168.1.5,10.5.1.2,172.1.2.4,3.4.2.1 192.168.1.5

Salvar as informações obtidas para um ficheiro:
nmap 192.168.2.2 > output.txt
nmap -oN /path/to/filename 192.168.2.2
nmap -oN output.txt 192.168.2.2

A ferramenta zenmap é a versão gráfica do nmap
apt-get install zenmap

Firewall no Desktop com UFW

ufw status verbose
ufw enable

O ufw é tão simples que apenas o comando acima já é suficiente para garantir a segurança. Fica tudo aberto para saída e tudo fechado para entrada.

Mas para customizações seguem mais informações.

ufw allow 65522
ufw logging on

ufw allow http
ufw allow https

ufw status verbose

O ufw é um firewall nativo do Ubuntu, que é bem simples de implementar que é uma interface para o IPTables.
Tutorial - https://help.ubuntu.com/community/UFW

Verificando seu status:
sudo ufw status # Estado: inativo

Não requer instalação, pois ele já vem instalado por padrão no Ubuntu. Apenas precisamos habilitá-lo.

sudo ufw enable

Ao habilitar ele fecha todas as entradas e abre todas as saídas e habilita na inicialização do sistema:

sudo ufw status verbose

ufw allow from 177.14.224.188 to any port 65522

Veja o que diz:
Estado: ativo
Logando: on (low)
Predefinido: deny (entrada), allow (saída), disabled (roteado)
Novos perfis: skip

Assim ninguém tem acesso a este servidor através da rede, nem pela web (porta 80), nem via ssh, nem ao banco de dados. Somente eu poderia acessar se fosse diretamente/fisicamente frente a ele ou então através do console, no caso do DigitalOcean.
Então precisamos abrir inicialmente a porta 22 para garantir o acesso. Depois trocaremos esta porta para que fique mais trabalhoso o acesso.

Como liberar uma portas ou serviços?
sudo ufw allow ssh
ou
sudo ufw allow 22
Vejamos:
sudo ufw status verbose:
nomeuser@ribaln ~ $ sudo ufw status verbose
Para Ação De
--- ---- --
22 ALLOW IN Anywhere
22 (v6) ALLOW IN Anywhere (v6)
Ele liberou a porta 22.

E se quisermos bloquear a porta 22?
sudo ufw deny 22

E se quiser remover esta regra deny que sempre aparece no status:
sudo ufw delete deny 22

Habilitando logs
ufw logging on

Agora eu quero que somente certo IP possa se conectar ao meu servidor
sudo ufw allow from 207.46.232.182

Agora que somente uma certe rede possa se conectar:
sudo ufw allow from 192.168.1.0/24

Agora que seja aberta a todos mas que via ssh somente para certo IP:
ufw allow from 192.168.0.4 to any port 22

Esta regra é indicada para maior segurança. Mesmo que use um desktop com internet ADSL, que muda o IP, mesmo assim. Quando seu IP mudar e você perder o acesso vá até o console da hospedagem e atualiza seu IP. Melhor ter um pouco mais de trabalho e manter seu servidor no ar.

Não devo ter uma regra permitindo que todos acessem a porta 22
E em seguida uma dizendo que somente um IP pode acessar a porta 22
Não vai funcionar pois a primeira regra está liberando todos.
Negar acesso a certo IP
sudo ufw deny from <ip address>

Exemplificando um servidor LAMP na sua DMZ:
ufw allow proto tcp from 192.168.5.0/24 to 192.168.100.2 port 22
ufw allow proto tcp from any to any port 80
ufw enable

Onde:

192.168.5.0/24 é sua rede interna.

192.168.100.2 é o IP interno do seu LAMP server

Ou um servidor de DNS apenas com ip válido:

ufw allow proto tcp from 200.200.200.201 to 200.200.200.10 port 22
ufw allow proto udp from any to 200.200.200.10 port 53
ufw allow proto tcp from 200.1.1.200 to 200.200.200.10 port 53
ufw enable

Onde:

200.200.200.201 é o IP nateado da sua rede
200.200.200.10 é o IP do seu servidor de DNS
200.1.1.200 é o seu DNS Slave

Monitorando Logs

tail -f /var/log/secure

tail -f /var/log/messages

Monitorar quem está tentando acessar o servidor em tempo real
tcpdump -n -e -ttt -i pflog0

grep CRON /var/log/syslog

service rsyslog restart

grep -i cron /var/log/syslog|tail -2

tail -fn 50 /var/log/apache2/error.log

Monitorando o Servidor

Caso encontre algo estranho no servidor nas pastas:
/tmp
/usr/local/sbin

A única maneira de recuperar um servidor atacado é:
- reinstalar o servidor do zero
- após a instalação executar o mtree e guardar o relatório
    mtree -x -ic -K cksum -K md5digest -K sha256digest -p / -X /home/ribafs/mtree-exclude > /tmp/mtree_primeiro.out
- realizar a atualizaçção do servidor e reboot
- ativar o firewall configurado e outros softwares que auxiliam a segurança
- testar o backup dos dados: sites, bancos, etc
- restaurar o backup
- torcer para que as falhas de segurança que permitiram o ataque tenham sido corrigidas
- após finalizar a instalação, configurações e recuperação doo backup e testes, executar novamente o mtree

Em caso de ataque fique à vontade para enviar um e-mail para FreeBSD-security@FreeBSD.org avisando. Descreva o que está acontecendo e porque você acredita que foi invadido.

Executando mtree

mtree -x -ic -K cksum -K md5digest -K sha256digest -p / -X /home/ribafs/mtree-exclude > /tmp/mtree_final.out

Podemos customizar este comando: rodando apenas em uma partição ou no disco inteiro, eliminando algum checksum, etc.

Compare os relatórios primeiro e final.

Forma simples:

mtree -f savedspec -f newspec > mtree.differences

Lembre que o FreeBSD te envia e-mail diariamente dando um status básico sobre o servidor.
Caso tenha alguma suspeita investique.

Ver o que está rodando

lsof -i (if installed)
netstat -an -f inet
ps -auxw | more
sockstat -4
fstat (with grep, read man page)

Monitorando a rede

nmap
nagios
snort
ntop

/etc/syslog.conf

Swatch
Monitorar arquivos modificados
find /var/www/html -type f -ctime -1 -exec ls -ls {} \;

Podemos colocar no cron para que seja executado a cada madrugada e nos envie um e-mail.

Procurar arquivos com 666
find /var/www/html -xdev -perm +o=w ! \(-type d -perm +o=t \) ! -type l -print

Procurar diretórios com 777
find /var/www/html -type d -perm -o+w -exec ls -ld {} \;

Procurar contas sem senha
awk -F: '($2 == "") {print}' /etc/shadow

Busca por backdoors

grep -iR 'c99' /usr/local/www/
grep -iR 'r57' /usr/local/www/
find /usr/local/www/ -name *.php -type f -print0 | xargs -0 grep c99

grep -RPn "(passthru|shell_exec|system|base64_decode|fopen|fclose|eval)" /usr/local/www/

Cheque por arquivos e diretórios ocultos
Outros exemplos de busca que ajudam a localizar exploits ou arquivos/pastas inesperadas:
find /home -type f | xargs grep -l MultiViews
find . -type f | xargs grep -l base64_encode <<< this can produce false positives, it is valid in many mail/graphics scripts
find . -type f | xargs grep -l error_reporting
. no Linux é o diretório atual
find / -name "[Bb]itch[xX]"
find / -name "psy*"
ls -lR | grep rwxrwxrwx > listing.txt

Monitorando o servidor

Memória RAM e swap
freecolor -om

Logs

/var/log/httpd-access.log
/var/log/httpd-error.log
/var/log/httpd-ssl_request.log
/var/log/messages.log
/var/log/pflog
/var/log/security

tcpdump -n -e -ttt -i pflog0
tcpdump -netttr /var/log/pflog -vv "tcp and port 80"
tcpdump -n -e -ttt -r /var/log/pflog
tcpdump -n -e -ttt -i pflog0
tcpdump -n -e -ttt -r /var/log/pflog port 80
tcpdump -n -e -ttt -r /var/log/pflog port 80 and host 192.168.1.3
tcpdump -n -e -ttt -i pflog0 host 192.168.4.2
tcpdump -n -e -ttt -i pflog0 inbound and action block and on wi0
This display the log, in real-time, of inbound packets that were blocked on the wi0 interface.

Ferramentas

Testes de vulnerabilidade online
https://geekflare.com/online-scan-website-security-vulnerabilities/

Bons clientes de sftp
FileZilla - http://filezilla.sourceforge.net/

WinSCP - http://winscp.net/
https://www.digitalocean.com/community/tutorials/an-introduction-to-securing-your-linux-vps

https://documentation.cpanel.net/display/68Docs/Configure+PHP+and+suEXEC
http://www.alain.knaff.lu/howto/PhpSuexec/
http://blog.stuartherbert.com/php/2008/01/18/using-suphp-to-secure-a-shared-server/
https://linsider.wordpress.com/2009/11/21/how-to-suphp-an-alternative-to-phpsuexec/
https://geekflare.com/nginx-webserver-security-hardening-guide/
https://geekflare.com/install-modsecurity-on-nginx/
https://geekflare.com/nginx-webserver-security-hardening-guide/
https://geekflare.com/modsecurity-owasp-core-rule-set-nginx/
https://www.vultr.com/docs/how-to-install-modsecurity-for-nginx-on-centos-7-debian-8-and-ubuntu-16-04
https://www.howtoforge.com/tutorial/install-nginx-with-mod_security-on-ubuntu-15-04/
https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-16-04
https://rikynity.wordpress.com/2012/05/30/installing-nginx-with-php5-and-php-fpm-and-mysql-support-on-ubuntu-11-04/
https://www.binarytides.com/install-nginx-php-fpm-mariadb-debian/

Tutorial: Nginx com PHP 7 e MySQL no Ubuntu 16.04 LTS
https://pplware.sapo.pt/tutoriais/tutorial-nginx-php-7-mysql-no-ubuntu-16-04-lts/?format=pdf

Programação

Quando programando para Joomla devemos utilizar seu framework, em especial suas funções de filtragem para bancos de dados e para limpeza de arquivos no sistema de arquivos, entre outras.

Não devemos esquecer as boas práticas de programação e manter o código bem organizado.

Devemos ter cuidado especial com todas as entradas de usuários: URLs, campos de formulários devem ser filtrados por caracteres especiais, especialmente campos hidden, cookies, etc para isso usando as funções do Joomla.

Devemos sempre usar criptografia em campos de senha, reforçar formulários com tokens. Devemos usar session para bloquear o acesso direto em todos os scripts.

No caso do Joomla, devemos ativar o recurso de URLs amigáveis para maior proteção contra os ataques via URL.

Nunca usar caminhos diretos em includes e sempre preferir require a include, pois os includes não param em erros nem disparam mensagens de erro.

Preferir algo como:
require_once(dirname(__FILE__) . '/../../../tiago.cfg');
Que ocultam o caminho real.

Visitar frequentemente sites públicos de divulgação de vulnerabilidades como o
Bons artigos e sites sobre vulnerabilidades no código:

https://docs.joomla.org/Archived:Vulnerable_Extensions_List
http://www.owasp.org/index.php/Main_Page
http://www.owasp.org/index.php/Category:Vulnerability
http://www.owasp.org/index.php/Category:OWASP_Joomla_Vulnerability_Scanner_Project
http://www.invasao.com.br/2009/01/31/vulnerabilidades-em-aplicacoes-web/
http://www.vivaolinux.com.br/dica/Explorando-vulnerabilidades-em-websites
http://www.vivaolinux.com.br/artigo/Vulnerabilidade-em-formulario-PHP/
http://www.portugal-a-programar.org/forum/index.php?topic=43795.0
http://samurai.intelguardians.com/
http://sectools.org/web-scanners.html
http://wiki.locaweb.com.br/pt-br/Verificando_vulnerabilidades_em_aplica%C3%A7%C3%B5es_Web
http://code.google.com/p/websecurify/
http://wiki.locaweb.com.br/pt-br/Joomla:_Aprenda_como_manter_o_seu_Website_seguro
http://www.criarweb.com/artigos/principais-vulnerabilidades-web.html

Quem usa Debian ou derivado:
apt-get install w3af

Ferramentas para Firefox
https://addons.mozilla.org/en-US/firefox/addon/161722/eula/88917?src=search (Acunetix)
https://addons.mozilla.org/en-US/firefox/addon/7597/eula/88410?src=search
https://addons.mozilla.org/firefox/downloads/file/77248/groundspeed-1.1-fx.xpi?src=search
https://addons.mozilla.org/firefox/downloads/file/101322/x-forwarded-for_spoofer-10.0.2-fx.xpi?src=search
https://addons.mozilla.org/en-US/firefox/addon/722/
https://addons.mozilla.org/en-US/firefox/addon/7598/eula/88414?src=search
https://addons.mozilla.org/en-US/firefox/addon/7595/eula/88415?src=search
https://addons.mozilla.org/en-US/firefox/addon/14598/eula/65681?src=search
https://addons.mozilla.org/en-US/firefox/addon/49858/eula/95600?src=search
https://addons.mozilla.org/en-US/firefox/addon/45607/eula/67793?src=search
https://addons.mozilla.org/en-US/firefox/addon/14600/eula/65683?src=search

http://github.com/codebutler/firesheep/downloads - varre sites a procura de sites sem senha e mostra conexões wi-fi sem senha permite visualizar uma lista de contas online que estão compartilhando a rede wi-fi aberta e basta um clique para se logar como usuário da conta. Dessa forma, as contas são invadidas facilmente.

O Firesheep oferece cookies a sites como Facebook e Twitter, que os utilizam para permitir o acesso do usuário - em lugar de solicitar senhas.

O próprio desenvolvedor da extensão, Eric Butler, expôs o ponto fraco da web afirmando que qualquer pessoa que visite um site inseguro conhecido pelo Firesheep terá nome e foto exibido em uma nova janela para todos os usuários conectados à rede.

Na lista de sites que permitem acesso por cookies estão Amaxon, Flickr, Google, WordPress, Yahoo e muitos outros.

A vulnerabilidade não é específica do Firefox e mudar de browser não ajuda. A melhor opção, pelo menos por enquanto, é evitar o uso de redes Wi-Fi abertas para acessar esse tipo de conteúdo.

E muitas outras que pode encontrar no site de addons da mozilla.

Referências

http://bsdadventures.com/harden-freebsd/print/
https://fleximus.org/howto/secure-freebsd
https://www.funzi.org/2015/03/01/basic-freebsd-hardening/
http://hemantthakur.github.io/2015/02/22/FreeBSD-Server-Hardening.html
https://linux-audit.com/freebsd-hardening-lynis/
https://geek.linuxman.pro.br/geek/ubuntu-pronto-para-guerra
https://www.thefanclub.co.za/how-to/how-secure-ubuntu-1604-lts-server-part-1-basics
https://linux-audit.com/ubuntu-server-hardening-guide-quick-and-secure/
https://hostpresto.com/community/tutorials/how-to-install-and-use-lynis-on-ubuntu-14-04/
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwjC5qTTl4DbAhXDGJAKHYj3C58QFjAAegQIABAs&url=http%3A%2F%2Fwww.blackhat.com%2Fpresentations%2Fbh-usa-02%2Fmurphey%2Fbh-us-02-murphey-freebsd.ppt&usg=AOvVaw0rnta7QakAT63JlA248HjP