

Construa seu próprio servidor de atualização do FreeBSD

Resumo

Este artigo descreve a construção de um Servidor Interno de Atualização do FreeBSD. O [freebsd-update-server](#) é escrito por [Colin Percival <cperciva@FreeBSD.org>](#), Oficial de Segurança Emérito do FreeBSD. Para usuários que acham conveniente atualizar seus sistemas em relação a um servidor de atualização oficial, a construção de seu próprio Servidor de Atualização do FreeBSD pode ajudar a estender sua funcionalidade ao suportar versões do FreeBSD ajustadas manualmente ou fornecer um espelho local que permitirá atualizações mais rápidas para vários computadores.

Índice

1. Agradecimentos	1
2. Introdução	1
3. Pré-requisitos	1
4. Configuração: Instalação e Setup	2
5. Compilando o Código de Atualização	4
6. Compilando um Patch	8
7. Dicas	12

1. Agradecimentos

Este artigo foi posteriormente impresso na revista [BSD Magazine](#).

2. Introdução

Usuários ou administradores experientes frequentemente são responsáveis por várias máquinas ou ambientes. Eles entendem as exigências e desafios difíceis de manter essa infraestrutura. Executar um Servidor de Atualização do FreeBSD torna mais fácil implantar patches de segurança e atualizações de software para máquinas de teste selecionadas antes de implementá-los nos servidores de produção. Também significa que vários sistemas podem ser atualizados a partir da rede local em vez de uma conexão com a Internet potencialmente mais lenta. Este artigo descreve os passos envolvidos na criação de um Servidor Interno de Atualização do FreeBSD.

3. Pré-requisitos

Para construir um Servidor Interno de Atualização do FreeBSD, alguns requisitos devem ser

atendidos.

- Um sistema FreeBSD em execução.



No mínimo, as atualizações requerem a criação de uma versão do FreeBSD maior ou igual a versão do release alvo para a distribuição.

- Uma conta de usuário com pelo menos 4 GB de espaço disponível. Isso permitirá a criação de atualizações para 7.1 e 7.2, mas os requisitos de espaço exatos podem mudar de versão para versão.
- Uma conta [ssh\(1\)](#) em uma máquina remota para fazer o upload das atualizações que serão distribuídas.
- Um servidor web, como o [Apache](#), com mais da metade do espaço necessário para a compilação. Por exemplo, as compilações de teste para 7.1 e 7.2 consomem uma quantidade total de 4 GB, e o espaço necessário no servidor web para distribuir essas atualizações é de 2,6 GB.
- Conhecimento básico de script shell com o shell Bourne, [sh\(1\)](#).

4. Configuração: Instalação e Setup

Faça o download do software [freebsd-update-server](#) instalando os pacotes [devel/git](#) e [security/ca_root_nss](#), e execute:

```
% git clone https://github.com/freebsd/freebsd-update-build.git freebsd-update-server
```

Atualize o arquivo `scripts/build.conf` adequadamente. Ele será usado durante todas as operações de compilação.

Aqui está o `build.conf` padrão, que deve ser modificado para atender ao seu ambiente.

```
# Arquivo de configuração principal para compilações do FreeBSD Update.
# Os dados de configuração específicos do lançamento estão mais abaixo
# na árvore de scripts.

# Local de onde buscar os releases
export FTP=ftp://ftp2.freebsd.org/pub/FreeBSD/releases ①

# Plataforma do host
export HOSTPLATFORM=`uname -m`

# Nome do host a ser usado dentro das jails
export BUILDHOSTNAME=${HOSTPLATFORM}-builder.daemonology.net ②

# Localização da chave privada SSH
export SSHKEY=/root/.ssh/id_dsa ③

# Conta SSH para a qual os arquivos são enviados
```

```
MASTERACCT=builder@wadham.daemonology.net ④
```

```
# Diretório para o qual os arquivos são enviados  
MASTERDIR=update-master.freebsd.org ⑤
```

Os parâmetros a serem considerados seriam:

- ① Este é o local de onde as imagens ISO são baixadas (pela sub-rotina `fetchiso()` do arquivo `scripts/build.subr`). A localização configurada não se limita a URIs FTP. Qualquer esquema de URI suportado pelo utilitário padrão `fetch(1)` deve funcionar bem. Personalizações para o código `fetchiso()` podem ser instaladas copiando o arquivo padrão `build.subr` para a área específica de release e arquitetura em `scripts/RELEASE/ARCHITECTURE/build.subr` e aplicando alterações locais.
- ② O nome do host de compilação. Essa informação será exibida em sistemas atualizados ao emitir o comando:

```
% uname -v
```

- ③ A chave SSH para enviar arquivos para o servidor de atualização. Um par de chaves pode ser criado digitando `ssh-keygen -t dsa`. Este parâmetro é opcional; a autenticação padrão por senha será usada como método de autenticação de fallback quando o `SSHKEY` não estiver definido. A página do manual `ssh-keygen(1)` contém informações mais detalhadas sobre o SSH e os passos apropriados para criar e usar uma chave.
- ④ Conta para enviar os arquivos para o servidor de atualização.
- ⑤ Diretório no servidor de atualização onde os arquivos são enviados.

O `build.conf` padrão fornecido com o código fonte do `freebsd-update-server` é adequado para compilar releases do FreeBSD para a arquitetura i386. Como exemplo de compilação de um servidor de atualização para outras arquiteturas, os seguintes passos resumem as alterações de configuração necessárias para a arquitetura amd64:

1. Crie um ambiente de compilação para o amd64:

```
% mkdir -p /usr/local/freebsd-update-server/scripts/7.2-RELEASE/amd64
```

2. Instale um arquivo `build.conf` no diretório de compilação recém-criado. As opções de configuração de compilação para o FreeBSD 7.2-RELEASE em amd64 devem ser semelhantes a:

```
# SHA256 hash of RELEASE disc1.iso image.  
export RELH=1ea1f6f652d7c5f5eab7ef9f8edbed50cb664b08ed761850f95f48e86cc71ef5 ①  
# Components of the world, source, and kernels  
export WORLDPARTS="base catpages dict doc games info manpages proflibs lib32"  
export SOURCEPARTS="base bin contrib crypto etc games gnu include krb5 \  
lib libexec release rescue sbin secure share sys tools \  
"
```

```
ubin usbin cddl"
export KERNELPARTS="generic"

# EOL date
export EOL=1275289200 ②
```

- ① A chave hash [sha256\(1\)](#) para o release desejado está publicada dentro do respectivo [anúncio de release](#).
- ② Para gerar o número "End of Life" para o arquivo build.conf, consulte o "Estimated EOL" publicado no [Site de Segurança do FreeBSD](#). O valor de `EOL` pode ser derivado da data listada no site da web, usando o utilitário [date\(1\)](#), por exemplo:

```
% date -j -f '%Y%m%d-%H%M%S' '20090401-000000' '+%s'
```

5. Compilando o Código de Atualização

O primeiro passo é executar o `scripts/make.sh`. Isso irá compilar alguns binários, criar diretórios e gerar uma chave de assinatura RSA usada para aprovar as compilações. Nesta etapa, uma frase secreta terá que ser fornecida para a criação final da chave de assinatura.

```
# sh scripts/make.sh
cc -O2 -fno-strict-aliasing -pipe findstamps.c -o findstamps
findstamps.c: In function 'usage':
findstamps.c:45: warning: incompatible implicit declaration of built-in function
'exit'
cc -O2 -fno-strict-aliasing -pipe unstamp.c -o unstamp
install findstamps ../bin
install unstamp ../bin
rm -f findstamps unstamp
Generating RSA private key, 4096 bit long modulus
.....++
.....++
e is 65537 (0x10001)

Public key fingerprint:
27ef53e48dc869eea6c3136091cc6ab8589f967559824779e855d58a2294de9e

Encrypting signing key for root
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
```



Anote a impressão digital da chave gerada. Este valor será necessário no `/etc/freebsd-update.conf` para as atualizações binárias.

Neste ponto, estamos prontos para preparar uma compilação.

```
# cd /usr/local/freebsd-update-server
# sh scripts/init.sh amd64 7.2-RELEASE
```

A seguir, segue um exemplo da execução de uma compilação *inicial*.

```
# sh scripts/init.sh amd64 7.2-RELEASE
Mon Aug 24 16:04:36 PDT 2009 Starting fetch for FreeBSD/amd64 7.2-RELEASE
/usr/local/freebsd-update-server/work/7.2-RELE100 of 588 MB 359 kBps 00m00s
Mon Aug 24 16:32:38 PDT 2009 Verifying disc1 hash for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 16:32:44 PDT 2009 Extracting components for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 16:34:05 PDT 2009 Constructing world+src image for FreeBSD/amd64 7.2-
RELEASE
Mon Aug 24 16:35:57 PDT 2009 Extracting world+src for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 23:36:24 UTC 2009 Building world for FreeBSD/amd64 7.2-RELEASE
Tue Aug 25 00:31:29 UTC 2009 Distributing world for FreeBSD/amd64 7.2-RELEASE
Tue Aug 25 00:32:36 UTC 2009 Building and distributing kernels for FreeBSD/amd64 7.2-
RELEASE
Tue Aug 25 00:44:44 UTC 2009 Constructing world components for FreeBSD/amd64 7.2-
RELEASE
Tue Aug 25 00:44:56 UTC 2009 Distributing source for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 17:46:18 PDT 2009 Moving components into staging area for FreeBSD/amd64
7.2-RELEASE
Mon Aug 24 17:46:33 PDT 2009 Identifying extra documentation for FreeBSD/amd64 7.2-
RELEASE
Mon Aug 24 17:47:13 PDT 2009 Extracting extra docs for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 17:47:18 PDT 2009 Indexing release for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 17:50:44 PDT 2009 Indexing world0 for FreeBSD/amd64 7.2-RELEASE
```

```
Files built but not released:
Files released but not built:
Files which differ by more than contents:
Files which differ between release and build:
kernel|generic|/GENERIC/hptrr.ko
kernel|generic|/GENERIC/kernel
src|sys|/sys/conf/newvers.sh
world|base|/boot/loader
world|base|/boot/pxeboot
world|base|/etc/mail/freebsd.cf
world|base|/etc/mail/freebsd.submit.cf
world|base|/etc/mail/sendmail.cf
world|base|/etc/mail/submit.cf
world|base|/lib/libcrypto.so.5
world|base|/usr/bin/ntpq
world|base|/usr/lib/libalias.a
world|base|/usr/lib/libalias_cuseeme.a
world|base|/usr/lib/libalias_dummy.a
world|base|/usr/lib/libalias_ftp.a
...
```

Em seguida, a compilação do target "world" é realizada novamente, com patches de "world". Uma explicação mais detalhada pode ser encontrada em `scripts/build.subr`.



Durante esse segundo ciclo de compilação, o daemon do protocolo de tempo de rede, `ntpd(8)`, é desativado. Conforme o [Colin Percival <cperciva@FreeBSD.org>](mailto:cperciva@FreeBSD.org), Oficial de Segurança Emérito do FreeBSD, "o código de compilação do `freebsd-update-server` precisa identificar carimbos de data/hora que são armazenados em arquivos para que eles possam ser ignorados ao comparar compilações e determinar quais arquivos precisam ser atualizados. Esse processo de localização de carimbos de data/hora funciona fazendo duas compilações com 400 dias de intervalo e comparando os resultados."

```
Mon Aug 24 17:54:07 PDT 2009 Extracting world+src for FreeBSD/amd64 7.2-RELEASE
Wed Sep 29 00:54:34 UTC 2010 Building world for FreeBSD/amd64 7.2-RELEASE
Wed Sep 29 01:49:42 UTC 2010 Distributing world for FreeBSD/amd64 7.2-RELEASE
Wed Sep 29 01:50:50 UTC 2010 Building and distributing kernels for FreeBSD/amd64 7.2-RELEASE
Wed Sep 29 02:02:56 UTC 2010 Constructing world components for FreeBSD/amd64 7.2-RELEASE
Wed Sep 29 02:03:08 UTC 2010 Distributing source for FreeBSD/amd64 7.2-RELEASE
Tue Sep 28 19:04:31 PDT 2010 Moving components into staging area for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 19:04:46 PDT 2009 Extracting extra docs for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 19:04:51 PDT 2009 Indexing world1 for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 19:08:04 PDT 2009 Locating build stamps for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 19:10:19 PDT 2009 Cleaning staging area for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 19:10:19 PDT 2009 Preparing to copy files into staging area for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 19:10:20 PDT 2009 Copying data files into staging area for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 12:16:57 PDT 2009 Copying metadata files into staging area for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 12:16:59 PDT 2009 Constructing metadata index and tag for FreeBSD/amd64 7.2-RELEASE
```

```
Files found which include build stamps:
kernel|generic|/GENERIC/hptrr.ko
kernel|generic|/GENERIC/kernel
world|base|/boot/loader
world|base|/boot/pxeboot
world|base|/etc/mail/freebsd.cf
world|base|/etc/mail/freebsd.submit.cf
world|base|/etc/mail/sendmail.cf
world|base|/etc/mail/submit.cf
world|base|/lib/libcrypto.so.5
world|base|/usr/bin/ntpq
world|base|/usr/include/osreldate.h
world|base|/usr/lib/libalias.a
world|base|/usr/lib/libalias_cuseeme.a
```

```
world|base|/usr/lib/libalias_dummy.a
world|base|/usr/lib/libalias_ftp.a
...
```

Por fim, a compilação é concluída.

```
Values of build stamps, excluding library archive headers:
v1.2 (Aug 25 2009 00:40:36)
v1.2 (Aug 25 2009 00:38:22)
@()FreeBSD 7.2-RELEASE 0: Tue Aug 25 00:38:29 UTC 2009
FreeBSD 7.2-RELEASE 0: Tue Aug 25 00:38:29 UTC 2009
    root@server.myhost.com:/usr/obj/usr/src/sys/GENERIC
7.2-RELEASE
Mon Aug 24 23:55:25 UTC 2009
Mon Aug 24 23:55:25 UTC 2009
    built by root@server.myhost.com on Tue Aug 25 00:16:15 UTC 2009
    built by root@server.myhost.com on Tue Aug 25 00:16:15 UTC 2009
    built by root@server.myhost.com on Tue Aug 25 00:16:15 UTC 2009
    built by root@server.myhost.com on Tue Aug 25 00:16:15 UTC 2009
Mon Aug 24 23:46:47 UTC 2009
ntpq 4.2.4p5-a Mon Aug 24 23:55:53 UTC 2009 (1)
 * Copyright (c) 1992-2009 The FreeBSD Project.
Mon Aug 24 23:46:47 UTC 2009
Mon Aug 24 23:55:40 UTC 2009
Aug 25 2009
ntpd 4.2.4p5-a Mon Aug 24 23:55:52 UTC 2009 (1)
ntpddate 4.2.4p5-a Mon Aug 24 23:55:53 UTC 2009 (1)
ntpdcc 4.2.4p5-a Mon Aug 24 23:55:53 UTC 2009 (1)
Tue Aug 25 00:21:21 UTC 2009
Tue Aug 25 00:21:21 UTC 2009
Tue Aug 25 00:21:21 UTC 2009
Mon Aug 24 23:46:47 UTC 2009

FreeBSD/amd64 7.2-RELEASE initialization build complete. Please
review the list of build stamps printed above to confirm that
they look sensible, then run
sh -e approve.sh amd64 7.2-RELEASE
to sign the release.
```

Aprove a compilação se tudo estiver correto. Mais informações sobre como determinar isso podem ser encontradas no arquivo de origem da distribuição chamado USAGE. Execute scripts/approve.sh, conforme indicado. Isso irá assinar a versão e mover os componentes para uma área de preparação adequada para o upload.

```
# cd /usr/local/freebsd-update-server
# sh scripts/mountkey.sh
```

```
# sh -e scripts/approve.sh amd64 7.2-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Signing build for FreeBSD/amd64 7.2-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Copying files to patch source directories for
FreeBSD/amd64 7.2-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Copying files to upload staging area for FreeBSD/amd64
7.2-RELEASE
Wed Aug 26 12:50:07 PDT 2009 Updating databases for FreeBSD/amd64 7.2-RELEASE
Wed Aug 26 12:50:07 PDT 2009 Cleaning staging area for FreeBSD/amd64 7.2-RELEASE
```

Após a conclusão do processo de aprovação, o procedimento de upload pode ser iniciado.

```
# cd /usr/local/freebsd-update-server
# sh scripts/upload.sh amd64 7.2-RELEASE
```



Caso o código de atualização precise ser reenviado, isso pode ser feito alterando para o diretório de distribuições públicas do release de destino e atualizando os atributos do arquivo já *enviado*.

```
# cd /usr/local/freebsd-update-server/pub/7.2-RELEASE/amd64
# touch -t 2008010101.01 uploaded
```

Os arquivos enviados precisarão estar no diretório raiz de documentos do servidor web para que as atualizações possam ser distribuídas. A configuração exata variará dependendo do servidor web utilizado. Para o servidor web Apache, consulte a seção [Configuração de servidores Apache](#) no Handbook.

Atualize o `KeyPrint` e o `ServerName` do cliente em `/etc/freebsd-update.conf`, e execute as atualizações conforme instruído na seção [Atualização do FreeBSD](#) do Handbook.



Para que o Servidor de Atualização do FreeBSD funcione corretamente, as atualizações tanto para o release *atual* quanto para o release *para o qual se deseja atualizar* precisam ser compiladas. Isso é necessário para determinar as diferenças nos arquivos entre os releases. Por exemplo, ao atualizar um sistema FreeBSD da versão 7.1-RELEASE para a versão 7.2-RELEASE, as atualizações precisarão ser compiladas e enviadas para o seu servidor de distribuição para ambas as versões.

Para referência, toda a execução do `init.sh` está anexada.

6. Compilando um Patch

Toda vez que uma [recomendação de segurança](#) ou um [aviso de segurança](#) é anunciado, uma atualização de patch pode ser compilada.

Para este exemplo, o 7.1-RELEASE será usado.

Algumas suposições são feitas para compilar uma release diferente:

- Configure a estrutura de diretórios correta para a compilação inicial.
- Execute uma compilação inicial para o 7.1-RELEASE.

Crie o diretório de patches do respectivo release em `/usr/local/freebsd-update-server/patches/`.

```
% mkdir -p /usr/local/freebsd-update-server/patches/7.1-RELEASE/  
% cd /usr/local/freebsd-update-server/patches/7.1-RELEASE
```

Como exemplo, considere o patch para o [named\(8\)](#). Leia o aviso e pegue o arquivo necessário em [Avisos de Segurança do FreeBSD](#). Mais informações sobre a interpretação do aviso podem ser encontradas no [Handbook do FreeBSD](#).

No [informe de segurança](#), este aviso é chamado de `SA-09:12.bind`. Após o download do arquivo, é necessário renomeá-lo para um nível de patch apropriado. É sugerido manter isso consistente com os níveis de patch oficiais do FreeBSD, mas seu nome pode ser escolhido livremente. Para esta compilação, vamos seguir a prática atualmente estabelecida do FreeBSD e chamá-la de `p7`. Renomeie o arquivo:

```
% cd /usr/local/freebsd-update-server/patches/7.1-RELEASE/; mv bind.patch 7-SA-09:12.bind
```



Ao executar uma compilação de nível de patch, presume-se que os patches anteriores estão em vigor. Quando uma compilação de patch é executada, ela executará todos os patches contidos no diretório de patch.

Podem ser adicionados patches personalizados a qualquer compilação. Use o número zero ou qualquer outro número.



Cabe ao administrador do Servidor de Atualização do FreeBSD tomar as medidas apropriadas para verificar a autenticidade de cada patch.

Neste ponto, um *diff* está pronto para ser compilado. O software verifica primeiro se um `scripts/init.sh` foi executado no respectivo release antes de executar a compilação do *diff*.

```
# cd /usr/local/freebsd-update-server  
# sh scripts/diff.sh amd64 7.1-RELEASE 7
```

A seguir, segue um exemplo de execução de compilação *diferencial*.

```
# sh -e scripts/diff.sh amd64 7.1-RELEASE 7  
Wed Aug 26 10:09:59 PDT 2009 Extracting world+src for FreeBSD/amd64 7.1-RELEASE-p7  
Wed Aug 26 17:10:25 UTC 2009 Building world for FreeBSD/amd64 7.1-RELEASE-p7  
Wed Aug 26 18:05:11 UTC 2009 Distributing world for FreeBSD/amd64 7.1-RELEASE-p7
```

```
Wed Aug 26 18:06:16 UTC 2009 Building and distributing kernels for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 18:17:50 UTC 2009 Constructing world components for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 18:18:02 UTC 2009 Distributing source for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 11:19:23 PDT 2009 Moving components into staging area for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 11:19:37 PDT 2009 Extracting extra docs for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 11:19:42 PDT 2009 Indexing world0 for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 11:23:02 PDT 2009 Extracting world+src for FreeBSD/amd64 7.1-RELEASE-p7
Thu Sep 30 18:23:29 UTC 2010 Building world for FreeBSD/amd64 7.1-RELEASE-p7
Thu Sep 30 19:18:15 UTC 2010 Distributing world for FreeBSD/amd64 7.1-RELEASE-p7
Thu Sep 30 19:19:18 UTC 2010 Building and distributing kernels for FreeBSD/amd64 7.1-RELEASE-p7
Thu Sep 30 19:30:52 UTC 2010 Constructing world components for FreeBSD/amd64 7.1-RELEASE-p7
Thu Sep 30 19:31:03 UTC 2010 Distributing source for FreeBSD/amd64 7.1-RELEASE-p7
Thu Sep 30 12:32:25 PDT 2010 Moving components into staging area for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:32:39 PDT 2009 Extracting extra docs for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:32:43 PDT 2009 Indexing world1 for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:35:54 PDT 2009 Locating build stamps for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:36:58 PDT 2009 Reverting changes due to build stamps for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:37:14 PDT 2009 Cleaning staging area for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:37:14 PDT 2009 Preparing to copy files into staging area for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:37:15 PDT 2009 Copying data files into staging area for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:43:23 PDT 2009 Copying metadata files into staging area for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:43:25 PDT 2009 Constructing metadata index and tag for FreeBSD/amd64 7.1-RELEASE-p7
...
Files found which include build stamps:
kernel|generic|/GENERIC/hptrr.ko
kernel|generic|/GENERIC/kernel
world|base|/boot/loader
world|base|/boot/pxeboot
world|base|/etc/mail/freebsd.cf
world|base|/etc/mail/freebsd.submit.cf
world|base|/etc/mail/sendmail.cf
world|base|/etc/mail/submit.cf
world|base|/lib/libcrypto.so.5
world|base|/usr/bin/ntpq
world|base|/usr/include/osreldate.h
world|base|/usr/lib/libalias.a
world|base|/usr/lib/libalias_cuseeme.a
world|base|/usr/lib/libalias_dummy.a
world|base|/usr/lib/libalias_ftp.a
...
```

```

Values of build stamps, excluding library archive headers:
v1.2 (Aug 26 2009 18:13:46)
v1.2 (Aug 26 2009 18:11:44)
@()FreeBSD 7.1-RELEASE-p7 0: Wed Aug 26 18:11:50 UTC 2009
FreeBSD 7.1-RELEASE-p7 0: Wed Aug 26 18:11:50 UTC 2009
    root@server.myhost.com:/usr/obj/usr/src/sys/GENERIC
7.1-RELEASE-p7
Wed Aug 26 17:29:15 UTC 2009
Wed Aug 26 17:29:15 UTC 2009
    built by root@server.myhost.com on Wed Aug 26 17:49:58 UTC 2009
    built by root@server.myhost.com on Wed Aug 26 17:49:58 UTC 2009
    built by root@server.myhost.com on Wed Aug 26 17:49:58 UTC 2009
    built by root@server.myhost.com on Wed Aug 26 17:49:58 UTC 2009
Wed Aug 26 17:20:39 UTC 2009
ntpq 4.2.4p5-a Wed Aug 26 17:29:42 UTC 2009 (1)
    * Copyright (c) 1992-2009 The FreeBSD Project.
Wed Aug 26 17:20:39 UTC 2009
Wed Aug 26 17:29:30 UTC 2009
Aug 26 2009
ntpd 4.2.4p5-a Wed Aug 26 17:29:41 UTC 2009 (1)
ntpddate 4.2.4p5-a Wed Aug 26 17:29:42 UTC 2009 (1)
ntpdc 4.2.4p5-a Wed Aug 26 17:29:42 UTC 2009 (1)
Wed Aug 26 17:55:02 UTC 2009
Wed Aug 26 17:55:02 UTC 2009
Wed Aug 26 17:55:02 UTC 2009
Wed Aug 26 17:20:39 UTC 2009
...

```

As atualizações são impressas e a aprovação é solicitada.

```

New updates:
kernel|generic|/GENERIC/kernel.symbols|f|0|0|0555|0|7c8dc176763f96ced0a57fc04e7c1b8d79
3f27e006dd13e0b499e1474ac47e10|
kernel|generic|/GENERIC/kernel|f|0|0|0555|0|33197e8cf15bbbac263d17f39c153c9d489348c2c5
34f7ca1120a1183dec67b1|
kernel|generic|/|d|0|0|0755|0||
src|base|/|d|0|0|0755|0||
src|bin|/|d|0|0|0755|0||
src|cddl|/|d|0|0|0755|0||
src|contrib|/contrib/bind9/bin/named/update.c|f|0|10000|0644|0|4d434abf0983df9bc474356
70d307fa882ef4b348ed8ca90928d250f42ea0757|
src|contrib|/contrib/bind9/lib/dns/opensslsa_link.c|f|0|10000|0644|0|c6805c39f3da2a06
dd3f163f26c314a4692d4cd9a2d929c0acc88d736324f550|
src|contrib|/contrib/bind9/lib/dns/opensslrsa_link.c|f|0|10000|0644|0|fa0f7417ee9da42c
c8d0fd96ad24e7a34125e05b5ae075bd6e3238f1c022a712|
...
FreeBSD/amd64 7.1-RELEASE update build complete. Please review
the list of build stamps printed above and the list of updated
files to confirm that they look sensible, then run
sh -e approve.sh amd64 7.1-RELEASE

```

to sign the build.

Siga o mesmo processo mencionado anteriormente para aprovar uma compilação:

```
# sh -e scripts/approve.sh amd64 7.1-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Signing build for FreeBSD/amd64 7.1-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Copying files to patch source directories for
FreeBSD/amd64 7.1-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Copying files to upload staging area for FreeBSD/amd64
7.1-RELEASE
Wed Aug 26 12:50:07 PDT 2009 Updating databases for FreeBSD/amd64 7.1-RELEASE
Wed Aug 26 12:50:07 PDT 2009 Cleaning staging area for FreeBSD/amd64 7.1-RELEASE
```

The FreeBSD/amd64 7.1-RELEASE update build has been signed and is ready to be uploaded. Remember to run

```
sh -e umountkey.sh
```

to unmount the decrypted key once you have finished signing all the new builds.

Depois de aprovar a compilação, faça o upload do software:

```
# cd /usr/local/freebsd-update-server
# sh scripts/upload.sh amd64 7.1-RELEASE
```

Para referência, toda a execução do [diff.sh](#) está anexada.

7. Dicas

- Se um release personalizado for compilado usando o [procedimento](#) nativo `make release`, o código do `freebsd-update-server` funcionará a partir do seu release. Como exemplo, um release sem ports ou documentação pode ser criado limpando a funcionalidade referente às sub-rotinas de documentação `findextradocs ()`, `addextradocs ()` e alterando o local de download em `fetchiso ()`, respectivamente, em `scripts/build.subr`. Como último passo, altere a chave [sha256\(1\)](#) em `build.conf` no seu respectivo release e arquitetura e você estará pronto para compilar o seu release personalizado.

```
# Compare ${WORKDIR}/release and ${WORKDIR}/$1, identify which parts
# of the world|doc subcomponent are missing from the latter, and
# build a tarball out of them.
findextradocs () {
}
# Add extra docs to ${WORKDIR}/$1
addextradocs () {
}
```

- Adicionar flags `-j NUMERO` para os alvos `buildworld` e `obj` no script `scripts/build.subr` pode acelerar o processamento, dependendo do hardware utilizado, no entanto, não é necessário. Usar esses flags em outros alvos não é recomendado, pois pode fazer com que a compilação se torne pouco confiável.

```
# Build the world
log "Building world"
cd /usr/src &&
make -j 2 ${COMPATFLAGS} buildworld 2>&1
# Distribute the world
log "Distributing world"
cd /usr/src/release &&
make -j 2 obj &&
make ${COMPATFLAGS} release.1 release.2 2>&1
```

- Crie um registro [DNS SRV](#) apropriado para o servidor de atualização e coloque outros atrás dele com pesos variáveis. O uso dessa facilidade fornecerá espelhos de atualização, no entanto, essa dica não é necessária, a menos que você deseje fornecer um serviço redundante.

```
_http._tcp.update.myserver.com.    IN SRV    0 2 80   host1.myserver.com.
                                     IN SRV    0 1 80   host2.myserver.com.
                                     IN SRV    0 0 80   host3.myserver.com.
```