

Manual de instalação do OPENBUS 2.0.0

Tecgraf

6 de março de 2014

Sumário

1	Introdução	1
2	Instalação	2
3	Validadores	3
3.1	Para que servem?	3
3.2	Validadores LDAP	3
3.3	Validadores personalizados	4
4	Configurações opcionais para administradores do servidor	4
4.1	Inicialização	4
4.2	Monitoramento	5
4.3	Rotacionamento de logs	5
4.3.1	Rotacionamento de logs usando o logadm (plataforma Solaris)	5
5	FAQ	6
5.1	Não lembro a sintaxe do crontab. Qual é ela mesmo?	6
5.2	Não lembro como editar o agendamento do cron. Como é mesmo?	6
5.3	Onde devo salvar o arquivo de configuração do logrotate?	6
5.4	Precisa ser configurada alguma permissão especial? Qual o owner e o grupo que a configuração do logrotate precisa ter?	6
5.5	Onde os arquivos rotacionados são armazenados?	7
5.6	Preciso executar o comando “logrotate /etc/logrotate.d/openbus”?	7
5.7	Preciso configurar para o comando “logrotate /etc/logrotate.d/openbus” ser colocado na inicialização automática da máquina? Como saber que ele está executando?	7
5.8	Como usar o logrotate SEM precisar ter acesso de administrador na máquina?	7
5.9	Como instalar o logrotate como usuário comum numa Solaris 10?	7
5.10	Como configuro um validador personalizado?	8
5.11	Ainda estou com dúvidas. Como entro em contato?	8

1 Introdução

Este documento visa informar apenas o procedimento necessário para instalar um barramento OPENBUS [7] da versão 2.0.0. Caso tenha interesse de entender melhor o que é um barramento OPENBUS, consulte o seu manual de referência [8].

2 Instalação

O primeiro passo para instalar o barramento deve ser descompactar o pacote do barramento da plataforma desejada. Os pacotes estão disponibilizados no site oficial do projeto: <http://www.tecgraf.puc-rio.br/openbus>.

As plataformas oficialmente suportadas são:

- Linux Kernel 2.6 glibc 2.5 64 bits (Linux26g4_64)
- Linux Kernel 2.6 glibc 2.5 32 bits (Linux26g4)

Para outras plataformas geramos pacotes sob demanda, exemplos:

- Linux Kernel 3.2 glibc 2.11 64 bits (Linux32_64)
- Linux Kernel 3.2 glibc 2.11 32 bits (Linux32)
- Linux Kernel 2.4 glibc 2.3 64 bits (Linux24g3_64)
- Linux Kernel 2.4 glibc 2.3 32 bits (Linux24g3)
- Solaris 10 release 8/07 SPARC 64 bits (SunOS510_64)
- Solaris 10 release 8/07 SPARC 32 bits (SunOS510)

Os exemplos deste documento utilizam comandos do Bash shell (interpretador de linguagem de comandos) [1]. Para executar o barramento deve-se seguir o seguinte procedimento:

1. Extrair o pacote. Para fins de legibilidade, vamos guardar o caminho do local da extração em uma variável de ambiente. Porém, **essa declaração não é obrigatória!**

```
export OPENBUS_HOME=<local de extracao do pacote>
```

2. Testar se a execução do binário do barramento está funcionando

```
$OPENBUS_HOME/bin/busservices --help
```

3. Gerar um par de chaves com tamanho de 2048 bits para o barramento. A chave privada (arquivo com terminação *.key*) deve ser do formato PKCS8 codificada em DER, e o certificado (arquivo com terminação *.crt*) deve ser do formato X.509 codificado em DER. Esse par de chaves deve ser criado utilizando o comando *openssl* que acompanha o pacote.

```
export LD_LIBRARY_PATH="${OPENBUS_HOME}/lib:${LD_LIBRARY_PATH}"
```

```
# caso seja MacOS é preciso também:
```

```
export DYLD_LIBRARY_PATH="${OPENBUS_HOME}/lib:${DYLD_LIBRARY_PATH}"
```

```
export OPENSSL_HOME="${OPENBUS_HOME}/openssl"
```

```
${OPENBUS_HOME}/bin/openssl genrsa -out tmp_openssl.key 2048
```

```
${OPENBUS_HOME}/bin/openssl pkcs8 -topk8 -nocrypt \
```

```
-in tmp_openssl.key -out <nome do par de chaves>.key -outform DER
```

```
rm -f tmp_openssl.key
```

```
${OPENBUS_HOME}/bin/openssl req -config $OPENSSL_HOME/openssl.cnf -new -x509 \
```

```
-key <nome do par de chaves>.key -keyform DER \
```

```
-out <nome do par de chaves>.crt -outform DER
```

4. Configurar o validador (ver próxima seção) e executar o barramento (binário *busservices*), passando as configurações da chave privada e do validador a serem utilizados, as quais são obrigatórias para execução do OpenBus.

Maiores informações sobre como utilizar a configuração do *busservices* podem ser encontradas no manual de referência [8].

3 Validadores

3.1 Para que servem?

São responsáveis por autenticar o sistema/usuário que se conecta ao barramento. Atualmente pode-se utilizar 2 tipos de validadores no OpenBus: validadores LDAP ou validadores personalizados.

- Validadores LDAP - usam as informações contidas em um serviço de diretórios LDAP para autenticar o acesso de usuários ao barramento. Em ambientes corporativos, é comum a administração da rede utilizar servidores LDAP [6, 9] para a autenticação de usuários.
- Validadores personalizados - utilizam funções escritas em Lua para verificar a autenticidade dos usuários.

3.2 Validadores LDAP

O validador LDAP, disponibilizado em `openbus.core.services.passwordvalidator.LDAP`, tem suporte a servidores Microsoft Active Directory e OpenLDAP. As configurações do validador precisam estar no arquivo de configuração que é informado ao *busservices* através do parâmetro “-configs”. A seguir apresentamos quais são as propriedades, seus significados e possíveis valores:

ldap_servers indica uma lista de URLs de servidores LDAP. Cada URL deve seguir o formato `<protocol>://<server>:<port>`, onde `<protocol>` pode ser `ldaps` ou `ldap`. Exemplo:

```
ldap_servers = {
  "ldaps://server.mycompany.com",
  "ldap://otherserver.mycompany.com:389",
}
```

ldap_patterns indica uma lista de padrões de formação de nomes que serão usados na autenticação LDAP. Atualmente só há suporte para o padrão `%U`, e o validador irá substituir esse padrão pelo nome do usuário fornecido no ato do login. No exemplo abaixo apresentamos dois padrões, o primeiro é útil para autenticação em servidores Microsoft Active Directory (que utilizam UPN [3]) e o segundo é útil para autenticação em servidores OpenLDAP (que utilizam DN [4, 5]). Exemplo:

```
ldap_patterns = {
  "%U@project.mycompany.com",
  "cn=%U,ou=users,ou=groups,dc=company",
}
```

ldap_timeout indica um tempo máximo de espera (em segundos) da resposta do servidor LDAP. Exemplo:

```
ldap_timeout = 10
```

3.3 Validadores personalizados

Um validador personalizado precisa ser um módulo Lua [2] que deve retornar uma função que recebe como parâmetro uma tabela de configurações do barramento e retorna uma função `validator`. A função `validator` recebe como primeiro parâmetro o nome de usuário e, como segundo, a senha. Caso o par usuário/senha seja válido, deve-se retornar verdadeiro, caso contrário, falso. Um exemplo de validador que verifica se o nome do usuário é igual à senha é dado a seguir:

```
local function validator(name, password)
    if name == password then
        return true
    end
end

return function(configs) return validator end
```

Para ver um exemplo de como configurar um validador personalizado, consulte a seção FAQ 5.

4 Configurações opcionais para administradores do servidor

No pacote de instalação do barramento existem scripts que auxiliam algumas tarefas diárias comuns ao administrador da máquina onde o barramento está instalado. Esses scripts estão localizados na pasta *bin* e são eles:

bus-init script de inicialização para iniciar e parar o barramento, semelhante aos scripts do */etc/init.d* em servidores Unix.

bus-check-running script que verifica se o barramento está executando e, caso não esteja, inicia o barramento utilizando o script *bus-init*.

4.1 Inicialização

Recomendamos que o administrador coloque o script *bus-init* em */etc/init.d/openbus* (ou outro lugar semelhante em seu sistema operacional) e faça o link simbólico apropriado para cada *runlevel*. É **necessário editar o script *bus-init*** para redefinir o local da instalação do barramento, de modo que o script possa encontrar o executável do *busservices*.

Além disso, por padrão, este script salva um arquivo de nome *busservices.pid* (contendo o PID do processo do *busservices*) dentro do diretório de instalação do barramento. Caso o administrador queira, ele também pode mudar a localização desse arquivo dentro do script.

Para utilizar o script *bus-init* é **recomendado** que o administrador do sistema configure o barramento para utilizar um arquivo de configuração. Nesse caso, o administrador **precisa** decidir onde colocar esse arquivo de configuração. Há duas formas de fazer isso utilizando o *bus-init*:

1. Criando um arquivo */etc/default/openbus* e definindo a variável `OPENBUS_CONFIG`. Exemplo:

```
OPENBUS_CONFIG=/etc/openbus.cfg
export OPENBUS_CONFIG
```

2. Editando o script *bus-init* e alterando a variável `PARAMS` com o parâmetro “-configs” e a localização do arquivo de configuração desejado. Exemplo:

```
PARAMS=-configs /etc/openbus.cfg
```

4.2 Monitoramento

O script *bus-check-running* foi criado para ser utilizado em conjunto com o agendador de tarefas do sistema (comando *cron* no Unix). Este script **aceita dois parâmetros** sendo que apenas o primeiro é obrigatório. O primeiro argumento deve indicar o local de instalação do barramento e o segundo um destinatário de email para receber as notificações de reinício do barramento (caso não seja informado, será enviado email para root@hostname.). É importante observar que este script **depende** do script *bus-init*.

Além disso, caso o administrador coloque o *bus-init* em outro local (que não seja */etc/init.d/openbus*), **será necessário editar o script *bus-check-running*** para redefinir a variável *OPENBUS_INIT*. Da mesma forma, caso o administrador altere o local do arquivo *busservices.pid* **também será necessário editar este script** para redefinir a variável *PID_BUSSERVICES*.

Um exemplo de agendamento no *cron* para utilizar o *bus-check-running* é dado a seguir, neste exemplo consideramos que o barramento está instalado em */opt/openbus-2.0*:

```
*/10 * * * * /opt/openbus-2.0/bin/bus-check-running /opt/openbus-2.0
```

4.3 Rotacionamento de logs

Por padrão, o barramento imprime logs em tela. É **fortemente recomendado** configurar o barramento para salvar esses logs em arquivos, conforme documentado em [8]. Por isso, recomendamos uma estratégia para rotacionar os arquivos de log.

O comando **logrotate** é um utilitário para simplificar a administração de arquivos de logs bem comum na maioria das instalações Linux e Solaris. O logrotate permite a rotação automática, a compressão dos logs antigos, a remoção de logs muito velhos e mesmo o envio de emails contendo os arquivos de logs.

Para rotacionar os logs do barramento, basta configurar o logrotate com as linhas a seguir. Neste exemplo consideramos que o barramento salva o log em */var/log/openbus.log*:

```
/var/log/openbus.log {  
    weekly  
    compress  
    copytruncate  
    rotate 4  
}
```

Os significados das opções recomendadas acima são dados a seguir:

1. O rotacionamento dos logs ocorre **semanalmente**.
2. Os logs antigos são comprimidos.
3. A opção **copytruncate** é necessária, pois o barramento escreve continuamente no log e ocorrerá erro de escrita em disco, caso os descritores de arquivos sejam alterados. Essa opção copia os arquivos de logs e, depois, reseta o arquivo original. É o procedimento mais seguro nesses casos.
4. Armazena-se até **4** volumes dos logs antigos.

4.3.1 Rotacionamento de logs usando o logadm (plataforma Solaris)

O **logadm** é muito parecido com o logrotate. Para a rotação dos logs, nós iremos utilizar o *crontab* e o *logadm*. Caso não seja o root da máquina, você deve utilizar o parâmetro *-f* para informar um arquivo de configuração. Esse arquivo irá conter todos os logs que você deseja rotacionar. Abaixo temos um exemplo de como configurar o *logadmin* para rotacionar os logs do barramento.

- O agendamento do *cron* que faz o *logadm* ser executado todo dia às 5:00AM:

```
00 05 * * * /usr/sbin/logadm -f /etc/openbus-logadm.conf
```

- No arquivo `/etc/openbus-logadm.conf` temos:

```
/var/log/openbus.log -C 5 -P 'Mon Jul 26 20:13:01 2010' -c -p 7d -z 0
```

Os significados das opções recomendadas acima são dados a seguir:

1. Os parâmetros `-C 5 -c -p 7d -z 0` fazem com que o `logadm` guarde os 5 últimos logs, truncando-os e comprimindo-os no formato `.gz`. Essa ação será feita a cada 7 dias.
2. O parâmetro `-P` será adicionado pelo próprio `logadm` após a primeira execução, ele é responsável por controlar a próxima data que o rotacionamento ocorrerá.

5 FAQ

5.1 Não lembro a sintaxe do `crontab`. Qual é ela mesmo?

```
.----- minuto (0 - 59)
| .----- hora (0 - 23)
| | .----- dia do mês (1 - 31)
| | | .----- mês (1 - 12)
| | | | .---- dia da semana (0 - 6) (Domingo=0 ou 7)
| | | | |
* * * * * comando a ser executado e suas opções
```

5.2 Não lembro como editar o agendamento do `cron`. Como é mesmo?

Para adicionar um novo agendamento pode-se usar o arquivo global `/etc/crontab` (caso seja `root`) ou editar a tabela específica do usuário (todo usuário pode usar o `cron` para rodar tarefas periódicas). O comando é:

```
crontab -e
```

Um arquivo com os agendamentos já existentes será aberto e será possível adicionar conforme informado anteriormente. Se o arquivo está vazio é porque o usuário do sistema não tem nenhum agendamento ainda.

5.3 Onde devo salvar o arquivo de configuração do `logrotate`?

A maioria das instalações Linux recentes já dispõem do diretório `/etc/logrotate.d`. Nesses casos, basta criar um arquivo chamado `openbus` (pode ser outro nome de sua preferência) e salvar as configurações acima nele.

É importante que o administrador leia as páginas de manual do `logrotate` (`man logrotate`), pois podem haver diferenças na configuração para versões antigas desse utilitário. Uma das situações comuns é não haver suporte ao diretório `/etc/logrotate.d`. Nesses casos, basta adicionar as configurações acima no arquivo `/etc/logrotate.conf`.

5.4 Precisa ser configurada alguma permissão especial? Qual o `owner` e o grupo que a configuração do `logrotate` precisa ter?

Não é necessário nenhuma permissão especial, basta o proprietário ser `root` e ter leitura para qualquer usuário (pode haver plataformas onde o `logrotate` execute como outro usuário que não o `root`).

5.5 Onde os arquivos rotacionados são armazenados?

No mesmo diretório dos arquivos originais.

5.6 Preciso executar o comando “logrotate /etc/logrotate.d/openbus”?

Não é preciso. Normalmente o logrotate é cadastrado como uma tarefa diária no cron. Mas isso não impede de que o administrador execute tal comando passando o parâmetro force para obrigar o primeiro rotacionamento. Isso é indicado principalmente logo após a configuração inicial.

5.7 Preciso configurar para o comando “logrotate /etc/logrotate.d/openbus” ser colocado na inicialização automática da máquina? Como saber que ele está executando?

O logrotate não é um daemon, portanto não se mantém em execução. Normalmente, o logrotate é executado através do agendamento do cron. Caso não haja cron disponível em uma dada plataforma, então nesse caso será sim importante executar o comando do logrotate na inicialização da máquina.

5.8 Como usar o logrotate SEM precisar ter acesso de administrador na máquina?

O logrotate pode ser usado por usuários desprivilegiados (aqueles que não possuem acesso de root). Contudo, é muito comum que esses usuários ainda queiram rotacionar periodicamente seus arquivos de log. Nessa situação, recomendamos que o usuário adicione regras no agendador de tarefas do Unix (cron) para que o comando do logrotate seja chamado (ao menos) diariamente.

Para usuários desprivilegiados é importante usar alguns parâmetros do logrotate como o state. Esse parâmetro indica qual arquivo de estados será usado. O logrotate usa tal arquivo de estado para registrar a data da última rotação.

Segue um exemplo de agendamento do cron para executar o logrotate diariamente às 12:00 horas para o barramento instalado em /opt/openbus-2.0:

```
0 12 * * * /usr/sbin/logrotate --state /opt/openbus-2.0/logrotate.status /etc/openbus-logrotate.conf
```

É importante notar que o arquivo logrotate.status contendo o último estado da rotação foi mantido no diretório da instalação do barramento.

5.9 Como instalar o logrotate como usuário comum numa Solaris 10?

Nas plataformas Solaris a forma mais simples é baixar o pacote binário a partir do SunFreeware e usar diretamente da pasta do usuário. Um exemplo é dado a seguir:

1. Download e extração do pacote a partir do sunfreeware:

```
wget -c ftp://ftp.sunfreeware.com/pub/freeware/sparc/10/logrotate-3.7.6-sol10-sparc-local.gz
gunzip logrotate-3.7.6-sol10-sparc-local.gz
mkdir /tmp/install
pkgtrans logrotate-3.7.6-sol10-sparc-local /tmp/install
```

2. Entendendo o conteúdo do pacote e copiando para uma pasta pessoal o binário principal:

```
ls -l /tmp/install/SMClogr/reloc/
drwxr-xr-x  3 openbus  tecgraf    512 Apr 12 15:24 doc
drwxr-xr-x  3 openbus  tecgraf    512 Apr 12 15:24 man
```

```
drwxr-xr-x  2 openbus  tecgraf      512 Apr 12 15:24 sbin
cp /tmp/install/SMClogr/reloc/sbin/logrotate $HOME/bin
```

Após esse procedimento basta ter a pasta \$HOME/bin no PATH.

5.10 Como configuro um validador personalizado?

Um exemplo simples de configuração de um validador personalizado, seria criar um código como o apresentado em 3.3, e salvar o código em um arquivo com a extensão “.lua”. Depois, adicione o caminho onde foi salvo o arquivo à variável de ambiente LUA_PATH, da seguinte forma:

```
export LUA_PATH+=";<caminho onde está o validador>/?.lua"
```

Para maiores informações, consulte o manual da linguagem Lua [2].

5.11 Ainda estou com dúvidas. Como entro em contato?

Disponibilizamos uma lista pública de discussão com o intuito de reconhecer os usuários da nossa tecnologia, bem como receber sugestões e críticas que contribuam para a evolução do nosso projeto.

Maiores informações sobre a nossa lista de discussão veja <http://listas.tecgraf.puc-rio.br/mailman/listinfo/openbus-users>.

Referências

- [1] GNU. Bash reference manual. <http://www.gnu.org/software/bash/manual/bashref.html>, 2010.
- [2] Roberto Ierusalimsky. The programming language lua - modules. <http://www.lua.org/manual/5.1/manual.html#5.3>, February 2008.
- [3] Microsoft. UPN user principal name attribute in microsoft active directory. [http://msdn.microsoft.com/en-us/library/windows/desktop/ms680857\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms680857(v=vs.85).aspx), 2012.
- [4] OpenLDAP Foundation. DN distinguished name in ldap. <http://www.openldap.org/doc/admin24/intro.html#What%20is%20LDAP>, 2012.
- [5] OpenLDAP Foundation. RFC 4514 - lightweight directory access protocol (ldap): String representation of distinguished names. <http://www.ietf.org/rfc/rfc4514.txt>, 2012.
- [6] OpenLDAP Foundation. RFC 4510 - lightweight directory access protocol (ldap): Technical specification road map. <http://www.ietf.org/rfc/rfc4510.txt>, 2013.
- [7] TecGraf. OpenBus - Enterprise Integration Application Middleware. <http://www.tecgraf.puc-rio.br/openbus>, 2006.
- [8] TecGraf. *Manual do OpenBus 2.0.0*. TecGraf, 2012.
- [9] Wikipedia. Lightweight directory access protocol — Wikipedia, the free encyclopedia, 2012. [Online; accessed 06-March-2012].