BasicNews FAQ

http://www.viste-family.net/mateusz/software/basicnews/

Mateusz Viste

Last update: 12 Jul 2009

Table of Contents

How do I install BasicNews on my Linux box?

That's easy! Simply copy the executable file "basicnews" to your /sbin/ directory, the file “basicnews.cfg” to your /etc/, configure your superserver (inetd or xinetd) to listen on port TCP/119 and forward connections to /sbin/basicnews. Here's an example of a configuration for xinetd:

```
service nntp {
  disable = no
  socket_type = stream
  protocol = tcp
  wait = no
  user = root
  server = /sbin/basicnews
}
```

In many Linux distributions, it's enough to put a file containing the lines stated above to the /etc/xinetd.d/ directory, and restart xinetd using the “/etc/init.d/xinetd restart” command.

The equivalent configuration for inetd would be this single line, added to /etc/inetd.conf:

```
nntp    stream    tcp    nowait    root    /sbin/basicnews    basicnews
```

Note, that the example is using the root account to run the server. This is a good way to avoid any permissions-related troubles while testing, but it is definitely not a recommended setting for a running public server!

See the xinetd (or inetd) manual for details about the superserver's configuration.

How can I create some newsgroups on my BasicNews server?

Okay, the BasicNews thing is up and running. How can I create some newsgroups now? BasicNews will check your spool directory for any subdirectories. Each subdirectory will be used as a newsgroup. Therefore, to create groups, you will just have to create directories in your spool folder. The spool directory is “/var/basicnews/” by default (make sure that this directory exists!). For example, if you would like to create a group called, say, “my.local.newsgroup”, you could use the command “mkdir /var/basicnews/my.local.newsgroup”.

Important note: All groups must be written on disk in lower case, otherwise BasicNews won't find them!

Once the group is created, you may customize it a bit, creating a per-group configuration file in the spool directory, with the same name than the group, plus the “.conf” extension (for our example, it would be “/var/basicnews/my.local.newsgroup.conf”). This file allows to apply the following settings:

```
# Description of the group. That's the "long" description the server
# will return for your newsgroup.
Description=My private group

# Write protection - must be "1" if you want to send messages to the group,
# 0 will make the group read-only.
WriteAccess=1

# Authentication - Require (or not) authentication for this group.
# If set to "1", BasicNews will look at the basicnews.users file
# for credentials.
Auth=1

# Maximum article size. This setting allows you to set up a maximum
# size limit for any article posted to the newsgroup (note, that
# attachments are part of the article). If you would like to prevent
# users from sending articles bigger than, say, 512 KiB to your
# newsgroup, you will have to set "MaxArticleSize=524288" (the limit
# is set in bytes). Default is 0 (no limit).
MaxArticleSize=0
```

Importing existing spools

It is good to know, that you can easily import any existing NNTP spool directory into BasicNews. You'll just have to copy articles file to the group's subdirectory in BasicNews spool (obvioulsy, articles have to be stored in files, where each file has the name of the article number). Any overview data will be generated on the fly.

What about authentication?

BasicNews supports authentication. If you want to use it, you will have to create a file named “basicnews.users” in your spool directory. Then, you will have the choice to activate global authentication (the user will have to authenticate immediately after connecting to the server, so he won't be able for eg. to retrieve the list of available groups), or activate authentication on a per-group basis. The global setting is in /etc/basicnews.cfg, while the per-group activation has to be activated via the group's configuration file. As for the “basicnews.users” file, it may look like that:

```
# This is the list of users which may authenticate themselves on
# the BasicNews server.
#
# Each entry must have the following syntax:
# user:password
#
# Any line which begins by a "#" character is ignored.
john:somepassword
melinda:anotherpassword
karl:apassphrase
someanonymlogin:
debbie:debbiespassword
```

Note, that password-less logins are possible, simply by using the syntax “login:” (in the example above, the long “someanonymlogin” is a password-less login). For such logins, BasicNews won't ask for password, and the user will be authenticated immediately.
Once you configured the global behavior, you may choose to apply authentication per group, via the the group's configuration file (you just have to set the token “Auth” to “1”). Then, you will have to create a new user file for this specific group. For eg. for the “test.basic.news” group, the user file will be named “test.basic.news.users”, and will contain the list of users allowed to access this group. Here's an example of such file (access only for john and debbie):

```
## Group's user list ##
# This file contains the list of users
# allowed to access the given newsgroup.
#
# Any line which begins by a "#" character is ignored.
john
debbie
```

The NNTP authentication scheme is by nature vulnerable to brute-force attacks. The attacker would just have to try various combinations of logins and passwords the fastest he is able to.
BasicNews is aware of such nasty behavior, and apply a simple (but efficient) protection against this issue: any request asking for a protected content and coming with false credentials will be answered with a delay of 2 seconds. Note, that 2 seconds is a very long time to wait for a bruteforce attacker.

Security Warning

The password is passed from the client to the server in plain text across the network. Anyone listening somewhere in the packet's way with any variety of packet sniffer will be able to read the username and password in the clear as it goes across. And, in addition to that, the content itself is also going across the network in the clear, and so if the newsgroup contains any kind of sensitive information, the same packet sniffer would have access to that information as it went past, even if the username and password were not used to gain direct access to the newsgroup.

Is there any solution to these security risks? Yes! In fact, the big trouble here is the possibility to capture clear traffic directly from the wire. The ultimate solution is simply to encipher your NNTP traffic into SSL packets. That's what we commonly call “NNTPS” (that is, NNTP over SSL).
Read the NNTPS-related chapter of this manual to discover how to turn BasicNews into a secure NNTPS server.

Does BasicNews support NNTPS (NNTP over SSL)?

What is it all about?

Reading the title of this chapter, you probably though "Wow, does BasicNews really support SSL ciphering?". Well... the simple answer is "yes", although it is not technically true.
I won't explain here how does SSL works, nor how certificate are used to encipher and authenticate hosts, as these subjects are far beyond the scope of this FAQ. However, we will see how to create a SSL certificate using OpenSSL, and how to wrap BasicNews into an SSL communication.

First of all, we need to know (barely) what's the benefit of using SSL. We all know that NNTP communication is transferred as clear-text over the wires. It means that anybody who has a physical access to your wire, will be able to see what transit between you and the NNTP server you are connected to. Obviously, that includes any passwords you could provide to authenticate yourself. The solution to this problem is quite simple: use an end-to-end encryption mechanism, which will transport your NNTP traffic (in this case, the encryption mechanism will be SSL). Of course, if you don't care about that (eg. you host a news server with public accesses only), then you won't have much benefits from SSL (apart authenticating the server to avoid any possible spoofing, but again – it won't be discussed here). You probably already guessed, that NNTP encrypted over a SSL tunnel is what we commonly know as "NNTPS".

Here we are again: How do I turn BasicNews into a NNTPS server? As said before, BasicNews itself doesn't provide any SSL support, as it works on clear-text communication only. However, we can use any SSL wrapper to tunnel BasicNews through a secure (enciphered) channel. For the needs of this FAQ, we will discover how to configure a NNTPS server using BasicNews, the Stunnel (SSL) wrapper, and the OpenSSL suite in a Linux environment (really, it is not as hard as it sounds).

Install the Stunnel wrapper

There's no general guide-line for installing Stunnel on your server system. It will mostly depends of your Linux distribution. You may install it from sources, or apply the Stunnel package, or copy the required binaries, etc... If you're using a Debian Linux operating system, you can easily install the whole Stunnel package using the command below:

```
apt-get install stunnel
```

On other systems, you will probably have equivalent commands. Check your system's handbook for details.

Generate your own SSL certificate

SSL is based on a public key infrastructure, which means that we will need a pair of keys – a private one, and its public equivalent. Sorry for this techy language... it's not so important anyway - all you

have to remember, is that you will need a certificate file, which will contain your keys, plus some other informations (Diffie-Hellman parameters...). Before starting, you will have to check whether your system has an OpenSSL installation or not (it's enough to check if you get something after typing the "openssl" command from within your shell). If not, install it now.

To generate a valid SSL certificate containing both your private and public key, you can use a command of the following syntax:

```
openssl req -new -x509 -days 365 -nodes -config stunnel.cnf -out cert.pem
-keyout cert.pem
```

This creates a private key, and a self-signed certificate. Arguments mean:

-days 365	make this key valid for 1 year, after which it's not to be used anymore (you may want to increase this value)
-new	Generate a new key
-x509	Generate an X509 certificate (self sign)
-nodes	Don't put a password on this key (otherwise you would have to manually type the certificate's password at each Stunnel call)
-config stunnel.cnf	The OpenSSL configuration file to use (you will find it somewhere in your Stunnel installation)
-out cert.pem	Where to put the SSL certificate
-keyout cert.pem	Put the key in this file (keep the same than above)

The OpenSSL generator will ask you few questions, and one of them will be to give a "Common Name" (or CN) to your certificate. This is quite important, as this parameter may be checked by the client's NNTP software. The "Common Name" has to be exactly the machine's host name, as used by the final client to access your server (in most cases it will be its FQDN, like “news.mysecuresserver.net”, but it could be also its IP, if clients will access the server using its IP. If the certificate's Common Name would not match the FQDN used by the remote client to reach your site, the user's news reader could display a warning, telling that the certificate presented by your server is wrong.

Note, that Stunnel will often need some DH parameters, too. DH stands for "Diffie-Hellman", it is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel.
These DH parameters have to be in the same file than the server's certificate. To generate DH parameters for our freshly created certificate, we will use the following command:

```
openssl gendh 512 >> stunnel.pem
```

It may happen that your specific installation of Stunnel doesn't require DH parameters, but it won't harm to have them in the file anyway.

If you would like to know how to generate a more specific certificate (choose the algorithm, key length, etc..), go read the OpenSSL documentation.

The procedure explained above will provide you with a custom, self-signed certificate. Anyone can make a self-signed certificate. It is a totally valid SSL certificate. However, many SSL clients wish to verify the identity of the organization that signed the certificate. These SSL clients often have a hard-coded list of known organizations (Certificate Authorities) that sign keys after doing background checks, etc.

Since the key and certificate you just generated are not in the hard-coded list that your SSL client uses, you may get either an error or warning message when attempting to connect to your NNTPS server. If you wish to interact with third-party clients that have hard coded lists of acceptable Certificate Authorities, and you do not want annoying dialog boxes popping up for the user on the first (or all – depending of the client's implementation) connections, then yes, you will have to have your key signed by a valid Certificate Authority. Unfortunately, it won't be free. It won't be cheap either.

Configure the NNTPS access

Once you installed Stunnel, and generated an appropriate certificate for your server, you will have to configure your system to listen on a specific port and forward any incoming connections to BasicNews, passing first through the Stunnel wrapper. There are several ways to achieve that, I will present one of them, which is using the xinetd super-server.

In fact, configuring xinetd to use Stunnel is not harder than configuring a bare (NNTP) BasicNews server. The whole point is to tell xinetd to pass any streams coming to a specific port (usually, you will want to use the standard TCP/563 port) to STUNNEL, which will have to be instructed to forward the stream to BasicNews, assuring the encryption/decryption on the fly.

Here is a simple example of a xinetd section for a NNTPS service using Stunnel and BasicNews:

```
service nntps {
  disable = no
  socket_type = stream
  protocol = tcp
  wait = no
  user = news
  server = /usr/bin/stunnel
  server_args = -l /sbin/basicnews -p /etc/ssl/certs/stunnel.pem
  instances = 100
  per_source = 5
}
```

Of course, you may have to adjust it depending on your system's configuration. Important things here are “nntps”, “news” and (obviously) the paths to the Stunnel and Grumpy binaries, as well as your certificate file. “nntps” is the name of the service on which xinetd has to listen on (typically, “nntps” should be binded to the 563 port via your “/etc/services” file). “news” is the user which has to be used to run the Stunnel instance. For testing purpose, you could use the root user, although it is not recommended to run any public service with root rights!

If you would like to have more details, go read the documentation of xinetd and/or Stunnel.

Last thoughts

Note, that the BasicNews server does not contain any cryptography itself. However, please remember that import and/or export of cryptographic software, code providing hooks to cryptographic algorithms, and discussion about cryptography is illegal in some countries. It is imperative for you to know your local laws governing cryptography. I am not liable for anything you do that violates your local laws.