

ELog at BNL Physics Department

BV

[2014-07-18 Fri 11:07]

A few R&D groups, largely filled by members of the Electronic Detector Group in the BNL Physics department, recognized the need for an online logbook. A survey was done to find what was available and ELog was selected. This documents the selection, prototyping and production setup.

1 Comparison with other potential systems

ELog is by far the most popular system as judged by various Google searches. There is also significant, prior experience with ELog among the primary users of the new system. A few others were considered.

1.1 Fermilab ECL

ECL is a rewrite of a prior system from Fermilab called CRL but it had no expression of interest from the primaries.

1.2 Evernote

ELog and Evernote were compared.

1.3 Bloknot

After the Evernote comparison was done another option was found in our own backyard. This was bloknot written by our own Vlodya Tishchenko who uses it as his own personal logbook. A few observations of this system

- It has many similarities to ELog in content, presentation, and backend storage method
- It's extensible through plugins (PHP).

- It supports HTML, plain text and markdown

While it was recognized that bloknote was very impressive it was not selected largely due to the greater familiarity the primaries have with ELog and the lack of expected community/support for bloknote.

2 Access policy

The desired ELog access policy is:

- auth is required to read or write, no anonymous access
- read-write accounts are per-individual (no sharing)
- read-only accounts may be shared
- write-once, read always (no edits nor deletes by normal users)
- admin level access may delete entries
- otherwise, a flat access control, any read-write user can write to any logbook, any auth'd user can read any logbook

3 Logbook organization

The organization is to be a hierarchy of groups of logbooks. At top level are "WbLS" and "LAr" for these two R&D efforts. Future groups are expected. Organization below this top level is up to the group. One expected future expansion might be to add a "Summer Students" group with one logbook per student.

Initial groups

- WbLS
- LAr
- Compton Scattering

4 Installation

The `eelogd` process will run as a dedicated user "`eelog`" and will be monitored by `supervisord`. It will only listen on `localhost` and be exposed to the Internet via Apache (running as it's own user).

4.1 Prerequisite package installation

Assuming SL6.5.

Apache:

```
# yum install httpd mod_ssl crypto-utils
```

To build ELog from source:

```
# yum install git gcc openssl-devel
```

4.2 Prepare

Make the user to run the ELog daemon:

```
$ sudo useradd -c "ELog Server" -d /home/elog -m -r -s /bin/bash elog
$ sudo su - elog
```

4.3 Install ELog

```
$ mkdir code
$ cd code
$ git clone https://bitbucket.org/ritt/elog.git
$ git clone https://bitbucket.org/tmidas/mxml.git
$ cd elog
$ make ROOT=/home/elog/opt
$ make ROOT=/home/elog/opt install
```

4.4 Initial Configure

```
$ mkdir ~/cfg
$ cd ~/cfg
$ cp ~/code/elog/elogd.cfg .
$ git init
$ git add elogd.cfg
$ git commit -a -m "First commit of default config."
```

4.5 Test

```
$ ~/opt/usr/local/sbin/elogd -c ~/cfg/elogd.cfg -s ~/opt/usr/local/elog -d ~/logbooks
```

And, visit: <http://localhost:8080/>

4.6 Apache

In an Apache's SSL site config file's `VirtualHost` section, add:

```
Redirect permanent / http://localhost:8080/  
ProxyPass / http://localhost:8080/
```

ELog needs to be told what URL to assume:

```
[global]  
port = 8080  
URL = https://lycastus.phy.bnl.gov/elog/
```

4.7 Supervisor

The `supervisord` is used to keep Elog up and running. Since nothing useful comes with Red Hat linux, we install it from source:

```
$ cd ~/code  
$ wget https://pypi.python.org/packages/source/v/virtualenv/virtualenv-1.11.tar.gz  
$ tar -xvf virtualenv-1.11.tar.gz  
$ ./virtualenv-1.11/virtualenv  
$ source ./venv/bin/activate  
$ pip install supervisor  
$ emacs ~/cfg/supervisord.cfg ~/run-supervisord.sh
```

The config:

```
[supervisord]  
#nada  
[program:elog]  
command=/home/elog/opt/usr/local/sbin/elogd -v -c /home/elog/cfg/elogd.cfg -s /home/el
```

The run script:

```
#!/bin/bash  
source ~/code/venv/bin/activate  
supervisord -c ~/cfg/supervisord.cfg
```

Run it:

```
$ ~/run-supervisord.sh
```

Now, walk away.

5 Configuration

In the 4 section is the minimum to get ELog to answer requests through Apache/SSL. This section says how to configure ELog to implement the desired 2.

5.1 Devel Elog Config

The various human managed files for Elog is kept in the `gitosis@gateway.phy.bnl.gov:elog` git repository. This lets changes be tested on a devel server before pushed and pulled down to the production server.

Note, the `passwords` file is not version controlled as Elog updates it all the time.

Repeat the Elog installation. For testing Elog one does not need Apache nor `supervisord` setup. However, to keep the same paths in the configuration file, it is best to create an `elog` account and install things identically in `/home/elog` as above.

For the devel server:

```
elog$ git clone gitosis@gateway.phy.bnl.gov:elog
elog$ mv elog/* elog/.git* .
elog$ rmdir elog
elog$ ~/opt/usr/local/sbin/elogd -c ~/cfg/elogd.cfg -s ~/opt/usr/local/elog -n localhost
```

Important note: one must tweak the URL setting in the config file to match the testing server and the remember to return it before committing.

```
# production:
#URL = https://elog.phy.bnl.gov/
# testing:
URL = http://localhost:8080/
```

6 Security

6.1 Local file path leaks

ELog will leak local file system paths on 404. This makes BNL Nessus cranky. This patch comments out the offending bits.

```
diff --git a/src/elogd.c b/src/elogd.c
index 88b709f..277ba30 100755
```

```

--- a/src/elogd.c
+++ b/src/elogd.c
@@ -8195,13 +8195,11 @@ void send_file_direct(char *file_name)

    rsprintf("<body><h1>Not Found</h1>\r\n");
    rsprintf("The requested file <b>");
-   /*
    strencode2(encodedname, file_name, sizeof(encodedname));
    if (strchr(file_name, DIR_SEPARATOR))
rsprintf("%s", encodedname);
    else
rsprintf("%s%c%s", dir, DIR_SEPARATOR, encodedname);
-   */
    rsprintf("</b> was not found on this server<p>\r\n");
    rsprintf("<hr><address>ELOG version %s</address></body></html>\r\n\r\n", VERSIO
return_length = strlen_retbuf;

```