

Installing Volatility in Ubuntu 9.04

Table of Contents

Abstract.....	2
Python.....	2
Installing Volatility.....	2
Extracting the tarball.....	3
First steps to setting up Volatility	3
Editing the main volatility file	5
Changing the permissions.....	5
Basic walkthrough.....	7
Installing Plugins	8
Installing malfind and malfind2	8
Installing volrip and volreg.....	8
Installing Volatility Subversion.....	9

Abstract.

Volatility is an open source python based extensible framework that assists investigators whether they be forensic examiners or malware analysts. The framework assists them in the examination of physical memory dumps, crash dumps and hibernation files. Volatility currently has support for Windows XP memory images.

This guide will assist examiners in setting up Volatility on Ubuntu Linux. This guide may also be used to assist in the setup on other Linux variants.

Python

The Volatility framework is developed using python, which on the majority of distributions is supplied by default. Using python also means that it can be utilised on other Operating Systems including Microsoft Windows, or Apple's MAC OSX. For Microsoft Windows please refer to Gleeda's guide on setup¹. You can check the current version available by starting a terminal and typing `python -V`.

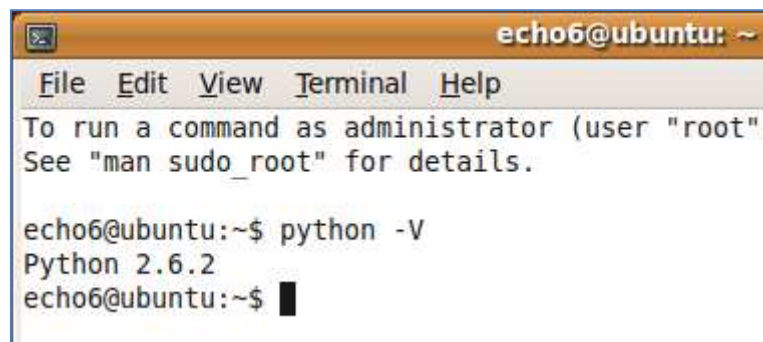
A screenshot of a terminal window titled 'echo6@ubuntu: ~'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', and 'Help'. Below the menu bar, there is a message: 'To run a command as administrator (user "root") See "man sudo_root" for details.' The terminal shows the command 'python -V' being entered and the output 'Python 2.6.2' being displayed. The prompt 'echo6@ubuntu:~\$' is visible before and after the command.

Figure 1: Displaying python version

Volatility provides support for "plugins" which further enhance the framework's capabilities. Some of these plugins are dependent on other packages, such as perl. This guide will also assist examiners in providing the necessary dependencies required by such plugins.

Installing Volatility

At the time of writing this guide the latest version available was Volatility-1.3_Beta. Volatility is available from <https://www.volatilesystems.com/default/volatility/> If you wish to install the cutting edge version refer to the Installing Volatility Subversion section.

From the terminal you can fetch this version by using the following;

```
wget https://www.volatilesystems.com/volatility/1.3/Volatility-1.3_Beta.tar.gz
```

¹ Windows install guide, <http://volatility.googlecode.com/files/install.pdf>

Download links are also provided for md5,sha1 and public key encryption signed files. To verify the .tar.gz (tarball) file using md5 download <https://www.volatilesystems.com/volatility/1.3/md5sum-1.3> We can view the contents from the terminal using the `cat` command and compare the md5 value using `md5sum`.

```
echo6@ubuntu:~$ cat md5sum-1.3
77d05a5e93ea77425379a306024b739b Volatility-1.3_Beta.tar.gz
c381f8756dd2f8fa0b5e000ff7bf85f4 Volatility-1.3_Beta.zip
echo6@ubuntu:~$ md5sum Volatility-1.3_Beta.tar.gz
77d05a5e93ea77425379a306024b739b Volatility-1.3_Beta.tar.gz
echo6@ubuntu:~$
```

Figure 2: Using md5sum to confirm file integrity

Aaron Walters' **public key** is also available together with a signed digest for each of the packages, you can use these to verify the integrity and authenticity of the packages using the `gpg` command.

First import Aaron Walter's public key to your **keyring** so that it can be used to verify the authenticity of the tarball or other files. `gpg --import awalters.asc`. We can then use the `gpg -verify` option to verify the files we have downloaded.

```
echo6@ubuntu:~$ gpg --verify Volatility-1.3_Beta.tar.gz.asc Volatility-1.3_Beta.
tar.gz
gpg: Signature made Fri 15 Aug 2008 09:57:14 AM PDT using DSA key ID 77933036
gpg: Good signature from "Aaron Walters <awalters@volatilesystems.com>"
gpg: WARNING: This key is not certified with a trusted signature!
gpg: There is no indication that the signature belongs to the owner.
Primary key fingerprint: 67F3 43B3 4E96 F071 66E4 A09D 7D84 1682 7793 3036
echo6@ubuntu:~$
```

Figure 3: Verifying package files authenticity

Extracting the tarball.

Before you extract the tarball archive first view its contents, we can do this using the `tar` command

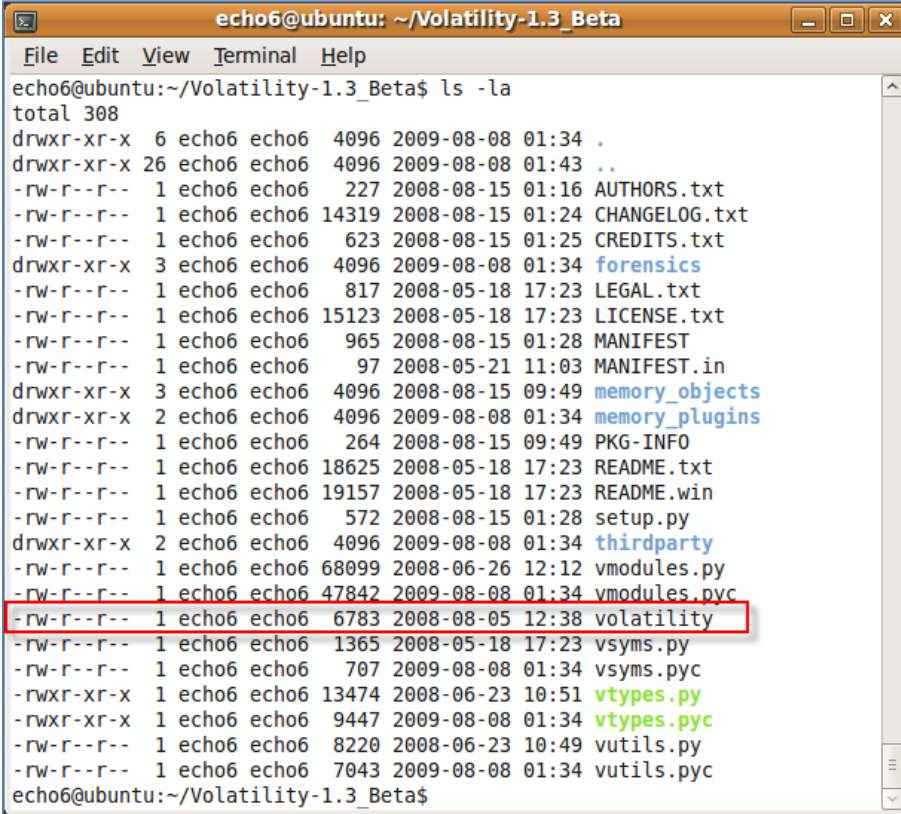
```
tar ztvf Volatility-1.3_Beta.tar.gz
```

By default this Volatility package creates its own directory called **Volatility-1.3_Beta** containing directories and other files required for its operation.

Now we use `tar zxvf Volatility-1.3_Beta.tar.gz` to extract the archive. If you have followed the guide without deviation the directory **Volatility-1.3_Beta** will be created in your users home directory, if you are unsure which directory you are currently at use the `pwd` command to print the current working directory.

First steps to setting up Volatility

We can now configure volatility for our environment. From the terminal change directory so that we are within the **Volatility-1.3_Beta** directory, `cd Volatility-1.3_Beta`. After entering `ls -l` you should see a listing of the directories and files as shown in figure 4.



```
echo6@ubuntu: ~/Volatility-1.3_Beta
File Edit View Terminal Help
echo6@ubuntu:~/Volatility-1.3_Beta$ ls -la
total 308
drwxr-xr-x  6 echo6 echo6  4096 2009-08-08 01:34 .
drwxr-xr-x 26 echo6 echo6  4096 2009-08-08 01:43 ..
-rw-r--r--  1 echo6 echo6   227 2008-08-15 01:16 AUTHORS.txt
-rw-r--r--  1 echo6 echo6 14319 2008-08-15 01:24 CHANGELOG.txt
-rw-r--r--  1 echo6 echo6   623 2008-08-15 01:25 CREDITS.txt
drwxr-xr-x  3 echo6 echo6  4096 2009-08-08 01:34 forensics
-rw-r--r--  1 echo6 echo6   817 2008-05-18 17:23 LEGAL.txt
-rw-r--r--  1 echo6 echo6 15123 2008-05-18 17:23 LICENSE.txt
-rw-r--r--  1 echo6 echo6   965 2008-08-15 01:28 MANIFEST
-rw-r--r--  1 echo6 echo6    97 2008-05-21 11:03 MANIFEST.in
drwxr-xr-x  3 echo6 echo6  4096 2008-08-15 09:49 memory_objects
drwxr-xr-x  2 echo6 echo6  4096 2009-08-08 01:34 memory_plugins
-rw-r--r--  1 echo6 echo6   264 2008-08-15 09:49 PKG-INFO
-rw-r--r--  1 echo6 echo6 18625 2008-05-18 17:23 README.txt
-rw-r--r--  1 echo6 echo6 19157 2008-05-18 17:23 README.win
-rw-r--r--  1 echo6 echo6   572 2008-08-15 01:28 setup.py
drwxr-xr-x  2 echo6 echo6  4096 2009-08-08 01:34 thirdparty
-rw-r--r--  1 echo6 echo6 68099 2008-06-26 12:12 vmodules.py
-rw-r--r--  1 echo6 echo6 47842 2009-08-08 01:34 vmodules.pyc
-rw-r--r--  1 echo6 echo6  6783 2008-08-05 12:38 volatility
-rw-r--r--  1 echo6 echo6  1365 2008-05-18 17:23 vsyms.py
-rw-r--r--  1 echo6 echo6   707 2009-08-08 01:34 vsyms.pyc
-rwxr-xr-x  1 echo6 echo6 13474 2008-06-23 10:51 vtypes.py
-rwxr-xr-x  1 echo6 echo6  9447 2009-08-08 01:34 vtypes.pyc
-rw-r--r--  1 echo6 echo6  8220 2008-06-23 10:49 vutils.py
-rw-r--r--  1 echo6 echo6  7043 2009-08-08 01:34 vutils.pyc
echo6@ubuntu:~/Volatility-1.3_Beta$
```

Figure 4: Listing Volatility files and directories

By default we can use the following to execute our newly installed volatility python application, i.e. `python volatility`. However we can also change it so that we simply need to supply `./volatility` to execute whilst within the Volatility-1.3_Beta directory, which is the way I prefer to have it setup.

To do this we need to edit the first line of the volatility python script, then change the permissions of the file to an executable. From the output of `ls -l` we can see that volatility has no execute permissions e.g. `-rw-r--r--` unlike the 'vtypes.py' file.

Editing the main volatility file

To edit the file we can either use **Gedit** GUI editor or the terminal editor **nano**.

The Gedit text editor is available within Gnome from **Application->Accessories->Text Editor**.

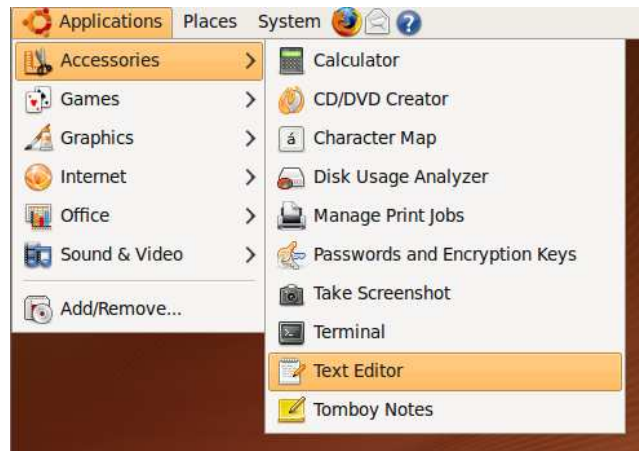


Figure 5: Accessing Gedit from Gnome

Edit the first line '`#!/c:\python\python.exe`' to read '`#!/usr/bin/env python`'. Note there is a space between **env** and **python**. As you can see the default setup is Windows centric, but we are using a far more versatile Operating System. There are some plugins which will only work under Linux, as we shall see later ;-). After editing the file save the changes. The change allows our Linux environment to identify the correct path to where python is installed. This should also allow other Linux environments to work with volatility and python.

Changing the permissions

To change the permissions of the **volatility** file, we use the command `chmod +x volatility`, we can confirm the permissions have been applied correctly using the `ls -l volatility` command.

```
echo6@ubuntu:~/Volatility-1.3_Beta$ ls -l volatility
-rwxr-xr-x 1 echo6 echo6 6783 2008-08-05 12:38 volatility
```

Figure 6: Checking volatility file permissions

Now when you do `./volatility` you should see the help output displayed in your terminal, as in Figure 7.

```

echo6@ubuntu:~/Volatility-1.3_Beta$ ./volatility
/home/echo6/Volatility-1.3_Beta/forensics/win32/crashdump.py:31: DeprecationWarning: the sha module is deprecated; use the hashlib module instead
  import sha

    Volatile Systems Volatility Framework v1.3
    Copyright (C) 2007,2008 Volatile Systems
    Copyright (C) 2007 Komoku, Inc.
    This is free software; see the source for copying conditions.
    There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR
A PARTICULAR PURPOSE.

    usage: volatility cmd [cmd_opts]

    Run command cmd with options cmd_opts
    For help on a specific command, run 'volatility cmd --help'

    Supported Internal Commands:
        connections      Print list of open connections
        connscan          Scan for connection objects
        connscan2         Scan for connection objects (New)
        datetime          Get date/time information for image
        dlllist           Print list of loaded dlls for each process
        dmp2raw           Convert a crash dump to a raw dump
        dmpchk            Dump crash dump information
        files             Print list of open files for each process
        hibinfo           Convert hibernation file to linear raw image
  
```

Figure 7: Help displayed in terminal output from ./volatility

The **DeprecationWarning** can be ignored, it will not affect volatility. If you wish to fix this you can edit the file `forensics/win32/crashdump.py` and edit line 31 to comment out the import that is no longer required using `#` at the start of that line. Hint: with **nano** if you use the key combination 'ctrl' + 'alt' + '_' together it will prompt for the line number.

```

GNU nano 2.0.9 File: forensics/win32/crashdump.py Modified
@contact:      awalters@volatilesystems.com,bdolangavitt@wesleyan.edu
@organization: Volatile Systems
"""
"""Tool: This tool generates a crash dump from a image of ram
"""
import os,optparse
import struct
#import sha
from time import gmtime, strftime
  
```

Figure 8: Commenting out import sha module

Basic walkthrough

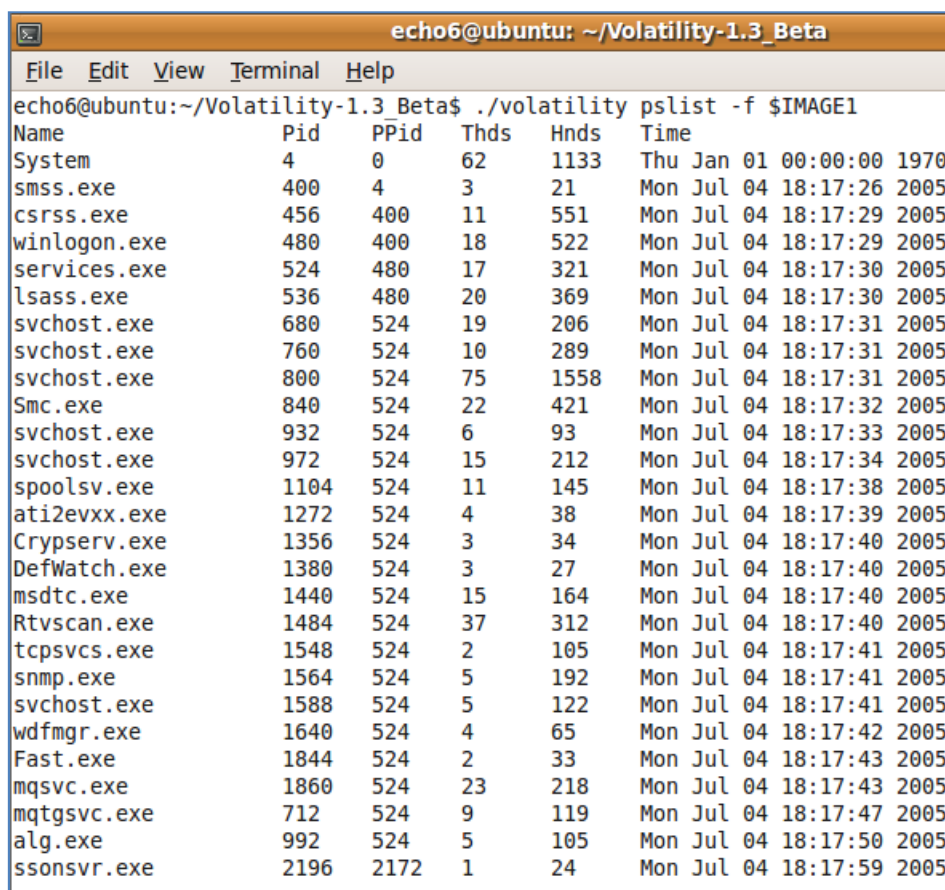
We can test our setup by downloading some sample images, we will use the ones available from the Computer Forensics Reference Data Sets project². Create a directory in your home directory to place the memory images, `mkdir ~/memdumps`, then change to that directory, `cd ~/memdumps` Download the archive of the first memory image.

```
wget http://www.cfreds.nist.gov/mem/memory-images.rar
```

Before we can extract the memory images from the archive we need to install `rar` using `sudo apt-get install rar`

Extract the archive `rar x memory-images.rar`. Now we can use Volatility with a suitable memory dump. Change back to your Volatility directory, `cd ~/Volatility-1.3_Beta`. Now use volatility to get a process listing from the memory dump file `xp-laptop-2005-07-04-1430.img`, `./volatility pslist -f ~/memdumps/xp-laptop-2005-07-04-1430.img`

A useful tip is to create an environment variable for our memory dump file, `IMAGE1=~/memdumps/xp-laptop-2005-07-04-1430.img`. Now rather than typing out the name of the directory and file in full we can use `./volatility -f pslist $IMAGE1`



Name	Pid	PPid	Thds	Hnds	Time
System	4	0	62	1133	Thu Jan 01 00:00:00 1970
smss.exe	400	4	3	21	Mon Jul 04 18:17:26 2005
csrss.exe	456	400	11	551	Mon Jul 04 18:17:29 2005
winlogon.exe	480	400	18	522	Mon Jul 04 18:17:29 2005
services.exe	524	480	17	321	Mon Jul 04 18:17:30 2005
lsass.exe	536	480	20	369	Mon Jul 04 18:17:30 2005
svchost.exe	680	524	19	206	Mon Jul 04 18:17:31 2005
svchost.exe	760	524	10	289	Mon Jul 04 18:17:31 2005
svchost.exe	800	524	75	1558	Mon Jul 04 18:17:31 2005
Smc.exe	840	524	22	421	Mon Jul 04 18:17:32 2005
svchost.exe	932	524	6	93	Mon Jul 04 18:17:33 2005
svchost.exe	972	524	15	212	Mon Jul 04 18:17:34 2005
spoolsv.exe	1104	524	11	145	Mon Jul 04 18:17:38 2005
ati2evxx.exe	1272	524	4	38	Mon Jul 04 18:17:39 2005
Crypserv.exe	1356	524	3	34	Mon Jul 04 18:17:40 2005
DefWatch.exe	1380	524	3	27	Mon Jul 04 18:17:40 2005
msdtc.exe	1440	524	15	164	Mon Jul 04 18:17:40 2005
Rtvscan.exe	1484	524	37	312	Mon Jul 04 18:17:40 2005
tcpsvcs.exe	1548	524	2	105	Mon Jul 04 18:17:41 2005
snmp.exe	1564	524	5	192	Mon Jul 04 18:17:41 2005
svchost.exe	1588	524	5	122	Mon Jul 04 18:17:41 2005
wdfmgr.exe	1640	524	4	65	Mon Jul 04 18:17:42 2005
Fast.exe	1844	524	2	33	Mon Jul 04 18:17:43 2005
mqsvc.exe	1860	524	23	218	Mon Jul 04 18:17:43 2005
mqtgsvc.exe	712	524	9	119	Mon Jul 04 18:17:47 2005
alg.exe	992	524	5	105	Mon Jul 04 18:17:50 2005
ssonsvr.exe	2196	2172	1	24	Mon Jul 04 18:17:59 2005

Figure 9: Using Volatility to list processes from a memory dump

² <http://www.cfreds.nist.gov/>

Installing Plugins

A list of plugins is maintained at the Forensicwiki³. The majority of these can simply be downloaded, unpacked if required, and placed into the Volatility-1.3_Beta directory or the Volatility-1.3_Beta/memory_plugins directory. Each time you add a plugin they should appear in the help output under "Supported Plugin Commands". You can get some additional help from each plugin, e.g. `./volatility malfind2 --help`

Installing malfind and malfind2

These plugins were written by Michael Hale Ligh available from his blog <http://mnin.blogspot.com>. We can download these two plugins directly into the memory_plugins directory.

```
cd ~Volatility-1.3_Beta/memory_plugins
wget http://mhl-malware-scripts.googlecode.com/files/malfind.py
wget http://mhl-malware-scripts.googlecode.com/files/malfind2.py
```

Before we can use them we have to install some dependencies. They rely on **pydasm**⁴ which is a python interface to **libdasm**⁵ a x86 disassembling library. It also relies on **pefile**⁶ which is a python module which reads and works with Portable Executable PE files.

Download and install

Fortunately pydasm is included with libdasm. An open source fork is available which is more up to date. This is the version we will install. This will require the use of the **subversion** tool. To install subversion use `sudo apt-get install subversion`. Now change to your home directory `cd ~` and use the following to download the latest code;

```
svn checkout http://libdasm.googlecode.com/svn/trunk/ svn/libdasm-read-only
```

Now enter the following series of commands;

```
cd svn/libdasm-read-only/
make
sudo make install
cd pydasm
sudo apt-get install python2.6-dev
python setup.py build_ext
sudo python setup.py install
```

pefile is easier to install because Ubuntu already has a package for it, `sudo apt-get install python-pefile`

Installing volrip and volreg

The packages volreg and volrip were written by Moyix (Brendan Dolvan-Gavitt), they utilise Harlan Carvey's excellent RegRipper⁷ tool to recover registry artifacts from a memory dump file. They provide the following plugins, **hivescan**, **hivelist**, **printkey**, **hashdump**, **lsadump** and **cachedump**⁸. A dependency is PyCrypto. This is installed using `sudo apt-get install python-crypto`.

³ ForensicWiki http://www.forensicswiki.org/wiki/List_of_Volatility_Plugins

⁴ pydasm, <http://dkbza.org/pydasm.html>

⁵ libdasm <http://www.nologin.org/main.pl?action=codeView&codeId=49>

⁶ pefile. <http://code.google.com/p/pefile/>

⁷ RegRipper <http://www.regripper.net/>

⁸ Registry plugins <http://moyix.blogspot.com/2009/01/memory-registry-tools.html>

Create a directory for your plugins, download the files to that directory and extract the tarballs to your Volatility directory. Use the following;

```
wget http://www.cc.gatech.edu/%7Ebrendan/volatility/dl/volreg-0.6.tar.gz
wget http://www.cc.gatech.edu/%7Ebrendan/volatility/dl/volrip-0.1.tar.gz
tar zcvf volreg-0.6.tar.gz -C ~/Volatility-1.3_Beta
tar zcvf volrip-0.1.tar.gz -C ~/Volatility-1.3_Beta
```

Installing Volatility Subversion

See Gleeda's guide⁹, which includes directions to installing the latest Volatility code under Linux.

Under Ubuntu the steps are straightforward though.

```
cd ~
sudo apt-get install subversion
svn checkout http://volatility.googlecode.com/svn/trunk svn
```

This will result in a directory being created in your home folder called `svn` which will contain a further directory called `volatility` which will contain the latest code. Repeat the previous steps to edit the main volatility file, change execution permissions and add additional plugins.

⁹ Windows SVN install, <http://volatility.googlecode.com/files/VolatilitySVN.pdf>