Tombuntu

News, Tips, and How-Tos for Ubuntu Linux

Simple File Encryption with OpenSSL

Published in December 12th, 2007
Posted by Tom in software

Linux has plenty of powerful encryption software, but what can you use if you just want to secure a couple files quickly? The OpenSSL toolkit works well for this. It comes installed with Ubuntu and can provide stronger encryption than you would ever need.

This is the basic command to encrypt a file:

```
openssl aes-256-cbc -a -salt -in secrets.txt -out secrets.txt.enc
```

How does this work?

- *openssl* is the command for the OpenSSL toolkit.
- *aes-256-cbc* is the encryption cipher to be used. (256bit AES is what the United States government uses to encrypt information at the Top Secret level.)
- *-a* means that the encrypted output will be base64 encoded, this allows you to view it in a text editor or paste it in an email. This is optional.
- *-salt* adds strength to the encryption and should always be used.
- *-in secrets.txt* specifies the input file.
- *-out secrets.txt.enc* specifies the output file.
- You will be prompted for a password.

It's not much use unless you can decrypted it:

```
openssl aes-256-cbc -d -a -in secrets.txt.enc -out secrets.txt.new
```

- *-d* decrypts data.
- *-a* tells OpenSSL that the encrypted data is in base64.

- *-in secrets.txt.enc* specifies the data to decrypt.
- *-out secrets.txt.new* specifies the file to put the decrypted data in.

Try out OpenSSL by decrypting this string (the password is *pass*):

```
U2FsdGVkX18YcWkbmhsN7M/MP1E+GLf4IqmNsa53T+A=
```

You can paste it into a text file and use the commands above, or use this command instead:

```
echo U2FsdGVkX18YcWkbmhsN7M/MP1E+GLf4IqmNsa53T+A= | openssl aes-256-cbc -d -a
```

See the OpenSSL man page for more detail on what it can do.

 Enjoyed this post? Subscribe to Tombuntu's RSS feed.

Share This

Related Posts

- Using TrueCrypt on Ubuntu for Encryption
- Create an Encrypted Private Directory with eCryptfs
- How to Use md5sum to Verify Data Integrity
- Public Key Authentication for SSH Made Easy
- Recover Deleted Files with Foremost

16 Responses to “Simple File Encryption with OpenSSL”

1. *mkwerner* says:
   December 12, 2007 at 9:42 am

   Hi Tom,
   I just wanted to let you know that I highly enjoy your tutorial posts. Like you, I started using Linux about 3 or 4 years ago, and I am by no means an expert! I recently switched to Kubuntu, and while there is a great deal of info on how to do various things, your format is easy to read and makes complete sense!

   Keep up the good work!

   Cheers,
   mkwerner

   Reply

2. *me* says:
   December 12, 2007 at 11:25 am

   thanks man, i enjoy your tutorials

   this encryption lesson is another one of my faves from your blog

   Reply
3. *Friday Linkfest « EdTechnology* says:
   December 14, 2007 at 7:41 am

   [...] over at tomubuntu.com, has a great refresher on using OpenSSL for encrypting and decrypting [...]

   Reply
4. *Ubuntu Life » Blog Archive » Encriptacion sencilla de ficheros con OpenSSL* says:
   December 14, 2007 at 10:20 am

   [...] Tombuntu me encuentro un tip muy interesante. Se trata de usar OpenSSL para encriptar nuestros ficheros de [...]

   Reply
5. *EncriptaciÃ³n sencilla de ficheros mediante OpenSSL « Xoanyeta's Weblog* says:
   January 25, 2008 at 6:55 pm

   [...] sencilla de ficheros mediante OpenSSL En Tombuntu me encuentro un tip muy interesante. Se trata de usar OpenSSL para encriptar nuestros ficheros de [...]

   Reply
6. *Anonymous* says:
   August 1, 2008 at 2:30 pm

   Great post! Good tip. Works nicely with bash too.

   Reply
7. *Create an Encrypted Private Directory with eCryptfs | Tombuntu* says:
   August 7, 2008 at 9:01 am

   [...] also written previously about simple file encryption with OpenSSL. Enjoyed this post? Subscribe to Tombuntu's RSS [...]

   Reply
8. *firmit* says:
   October 30, 2008 at 6:54 am

   As you say yourself "(…)[openSSL] can provide stronger encryption than you would ever need."

   True – so true 

   You also have other alternatives, escpecially 'bcrypt'. It uses the blowfish algorithm – more than good enough for simple file-encryption. Simply run

   $ bcrypt file

enter a passphrase, voila – file is encrypted to file.bfe. To decrypt, same command:

$ bcrypt file.bfe

Reply

9. *Yves* says:
November 13, 2008 at 5:30 am

Quick and clean encryption solution.
Thank you.

Reply

10. *Bookmarks for Monday, March 30th — Trevor Fitzgerald* says:
March 30, 2009 at 5:01 pm

[...] Simple File Encryption with OpenSSL | Tombuntu [...]

Reply

11. *[security] Encrypt files in the command line using OpenSSL « *NIX tricks* says:
October 3, 2009 at 1:39 pm

[...] Reference: link. [...]

Reply

12. *OpenSSL? ??? ??? ?? ??? ??* says:
October 12, 2009 at 8:27 am

[...] ??: http://tombuntu.com/index.php/2007/12/12/simple-file-encryption-with-openssl/ [...]

Reply

13. *Anthony thyssen* says:
October 28, 2009 at 7:16 pm

if you use a suffix such as '.enc' for encrypted files, you can also use VIM to edit an encrypted file. It asks for the password to decrypt then if you write it will ask for the password again to encrypt…

Add the following to your .vimrc file…

" Edit encrypted using openssl aes-256-cbc
augroup enc
autocmd!
autocmd BufReadPre,FileReadPre *.enc set bin
autocmd BufReadPre,FileReadPre *.enc set noswapfile
autocmd BufReadPost,FileReadPost *.enc set shell=sh
autocmd BufReadPost,FileReadPost *.enc set shellredir=>
autocmd BufReadPost,FileReadPost *.enc '[,']!openssl aes-256-cbc -d -a
autocmd BufReadPost,FileReadPost *.enc exe "doau BufReadPost ".expand("%:r")
autocmd BufReadPost,FileReadPost *.enc set nobin
autocmd BufReadPost,FileReadPost *.enc redraw!
autocmd BufWritePre,FileWritePre *.enc mark z
autocmd BufWritePre,FileWritePre *.enc set bin
autocmd BufWritePre,FileWritePre *.enc '[,']!openssl aes-256-cbc -a -salt
autocmd BufWritePost,FileWritePost *.enc undo

autocmd BufWritePost,FileWritePost *.enc set nobin
autocmd BufWritePost,FileWritePost *.enc ‘z
augroup END

To create a new file just start vim without a file name and then ‘write’ it to one ending in ‘.enc’ for example
:w secret_stuff.enc

To look at or modify the file just run vim secret_stuff.enc

WARNING: when saving make sure the file gets written correctly by looking at the output before quitting.

ASIDE: The above was developed from old PGP and GPG encrypted file techniques from vim.

Reply

14. *Anthony thyssen* says:
October 28, 2009 at 7:55 pm

Addendum — Remove the -a option from the above. When saving to a file there is no need to request ‘base64′ encoding! Save it directly as binary!

Reply

15. *Kevin* says:
April 30, 2010 at 10:50 pm

Very nice! I added this to my .profile

alias encrypt=”openssl aes-256-cbc -a -salt”
alias decrypt=”openssl aes-256-cbc -d -a -salt”

Reply

16. *#! » Simple bash variable security with OpenSSL* says:
November 17, 2010 at 5:25 pm

[...] standard we are using). The following example is cheerfully swiped from the example given over at Tombuntu. ?View Code BASH>openssl aes-256-cbc -a -salt -in config.plain -out [...]

Reply

Leave a Reply

Name

Mail (will not be published)

Website

Submit Comment

Subscribe

- 

Advertisements

Linode

- Tombuntu is hosted on a Linode virtual private server running Ubuntu 8.04 LTS Server Edition
  linode.com Linux Virtual Servers Notes from Setting Up Ubuntu Server on Linode

Top Posts

- How to Install Google Earth in Ubuntu 10.04 Remove Ubuntu Kernels You Don't Need Create a Bootable USB Drive the Easy Way in Ubuntu 8.10 Open RAR Archives in Ubuntu How to Install Docky in Ubuntu 9.10

Recent Entries

- How-to Run Unity in VirtualBox
- Ubuntu 11.04 Released
- Ubuntu 10.10 Released

- First Look at the Ubuntu Unity Desktop Environment
- Fix Volume Range Issue in PulseAudio
- Compiz Keyboard Shortcuts in Ubuntu 10.04
- Mark Shuttleworth Introduces Window Indicators
- Workarounds for Unrecognized Clicks in Flash Player
- How to Install Google Earth in Ubuntu 10.04
- Setting up Ubuntu 10.04 LTS

Categories

- Select Category

Archives

- Select Month