Scamp-5.5

Author:
Gerard Seibert

Released: 04/27/2013

IMPORTANT: FreeBSD users who install this program from the ports system should read the **'FreeBSD Note'** at the end of this document.

********** IMPORTANT **********
IMPORTANT: This is a "bash" script. It will probably fail if used under another shell.

If this is an update from an earlier version of this script, it is strongly that this script be run with the [-c] command line option to insure it is configured correctly. In addition, any existing configuration files should either be rebuilt using the "-e <name>" option or deleted and recreated.

The first time this script is initialized, it will run its basic configuration routine and exit. If you ever want to reconfigure the script, use the [-c] command line option to erase and create a default config file. Use [-C <name>] to create a new config section (after the default is created).
****

REQUIRED FILES:

This script was written using <Bash-4.2.42>. While it may work correctly with other scripting languages, there is no guarantee that it will.

1. Curl or Wget curl >= 7.24 recommended wget >= 1.14 recommended
   (a) `http://curl.haxx.se/`
   (b) `http://www.gnu.org/software/wget/`
2. rsync Version >= 3.0.9 recommended
   (a) `http://samba.anu.edu.au/rsync/`
3. gnupg Version >= 2.0.19 recommended
   (a) `http://lists.gnupg.org/pipermail/gnupg-announce/2009q1/000287.html`
   (b) `http://www.gnupg.org/`
4. clamav Version >= 0.97 recommended
   (a) `http://www.clamav.net/`

BASIC CONFIGURATION:

1. This script probably has to be run as ROOT or another privileged user who has proper READ/WRITE permissions to the "/etc" directory as well as the Clamav database.
2. This script has an interactive configuration function. You must run it from the console the first time you invoke it. After that, it will work fine from CRON.
3. The [-e] option will edit an existing config file. The file must be stated on the command line: scamp.sh -e <filename>
4. **Filename** is what you want to call the new config file. Use only letters, numbers and underscores. No special characters allowed. If the section does not exist, it will be created. The config-script will ask the same questions as when you first installed the script. When finished, it will exit. To use your new config section, just call it from the command line like you did when you created it. It will use the configuration file named. You should NOT give the configuration file a 'prefix'. It is not required.

   REMEMBER: The name is case sensitive!

5. The **"scamp.sh"** file is a BASH script for downloading and installing various **Clamav definition files.** The script will create a **"CONFIGURATION FILE"** when first run. The majority of the variables are 'hard coded' into the script. The only one that must be entered is the location of the Clamav database. See your **"clamd.conf"** file for the correct location.

   (a) Examples are:
      i. /var/db/clamav
      ii. /var/lib/clamav
      iii. /usr/local/share/clamav

6. You must enter the correct location or else the script will not work. The initialization will only occur the first time the script is run or when invoked by a command line option [-c].

COMMAND LINE OPTIONS:

- -c = Creates a new default config file and exits
- -C <filename> = Create a new config file section
- -D = Delete existing definition and configuration files.
- -e <filename> = Edit an existing config file
- -h = Usage screen.
- -l = Turns off the logging function
- -L = Turns on the logging function
- -q = Turns off printing of a summary screen (Error messages displayed)
- -Q = Turns on printing of a summary screen displayed.
- -r = Turns off the sleep function.
- -R = Turns on the random sleep timer. Between 0 & 9 minutes
- -v = Displays the script version and exits.

When available, lower case letter will turn an option off, while upper case will activate the function. Presently, only the 'log' and 'summary screen' and 'random download timer' functions are supported. They can be set permanently in the config file.

NOTE: "Random Download Timer"

The random download function is only useful when the script is run via **CRON**. It is ignored at other times. The function can be invoked via the command line using [-R]; i.e., **"scamp.sh -R"** for instance. You can save the setting permanently in the config file by running the script with the [-c] command line option and then answering the random download question with either 1 or 0. 0=off & 1=on.

EXIT CODES:

The following exit code values are available:

- 0 Success
- 1 Incorrect flag entered

- 2 No database specified
- 3 Unable to create required directory structure
- 4 Program must be run interactively
- 5 Unable to locate "which" binary
- 6 Missing binary: View error message for details
- 7 Error creating GPG file

INSTALLATION NOTES:

1. All of the Sanesecurity files and gpg keys are now keep in the Sanesecurity (sane) directory.
2. For the safest and most error free operation, I would recommend cleaning out the clamav database directory of all files and directories not installed by the 'freshclam' program itself. This appears to be even more important if you have been running another script to update the clamav database. Using the [-D] command line flag works fine for files installed using then 'scamp.sh' script. It is not guaranteed, and probably will not work, for files installed by other scripts.
3. That would probably include all BUT these files:
   (a) daily.cld
   (b) main.cld
   (c) mirrors.dat
   (d) stats.dat
4. Any directories should also be deleted.
5. Running the script for the first time after cleaning out the clamav database will insure a cleaner install of the new database files. Since it appears that different 'scripts' install a radically different configuration of definition files, this would also insure that only the ones installed by this script are made available to Clamav. It would also insure that outdated files are removed.

DOWNLOADED FILES:

The actual files download and installed by this script include the following.

The following databases are distributed and produced by Sanesecurity		
Database Name	Description	FP Risk
junk.ndb	General high hitting junk, containing spam/phishing/lottery/jobs/419s etc.	Low
jurlbl.ndb	Junk Url based	Low
jurlbla.ndb	Junk Url based autogenerated from various feeds	Med
lott.ndb	Lottery	Med
phish.ndb	Phishing	Low
rogue.hdb	Malware, Rogue anti-virus software and Fake codecs etc. Updated hourly to cover the latest malware threats.	Low

sanesecurity.ftm	Message file types (REQUIRED for best performance)	
sigwhitelist.ign2	Fast update file to whitelist any problem signatures. (REQUIRED 0.96rc1+)	
scam.ndb	Spam/scams	Low
spam.ldb	Spam detected using the new Logical Signature type	Med
spamimg.hdb	Spam images	Low
spamattach.hdb	Spam Spammed attachments such as pdf's/docs/rtf/zips	Low
spear.ndb	Spear phishing email addresses	Med
spearl.ndb	Spear phishing urls	Med
blurl.ndb	Blacklisted full urls over the last 7 days, covering malware/spam/phishing. URL's added only when main signatures have failed to detect but are known to be "bad".	Low
foxhole_generic.cdb	This database will block double extensions of certain common file formats that are contained within Zip/Rar and 7Zip files. These files will be detected only if they end in dangerous filestypes such as: pif, scr, exe, com, bat, cmd, vbs, lnk, cpl, vbs. Example signatures name formats: Sanesecurity.Foxhole.Zip_doc: blocks dangerous double extension .doc files, within zip files only Sanesecurity.Foxhole.Rar_xls: blocks dangerous double extension .xls files, within Rar files only Sanesecurity.Foxhole.Zip_hidden: blocks dangerous double extension files that are trying to hide their true extension, within zip files only	Med
foxhole_filename.cdb	This database will block certain commonly known malware filenames within Zip/Rar/7Zip files.	Med

foxhole_all.cdb	This database will block all files (single and double extensions) within Zip/Rar and 7Zip files that end in dangerous filestypes such as: pif, scr, exe, com, bat, cmd, vbs, lnk, cpl, vbs. This will be the most effective database of the three but also has the highest risk of false positives, unless you are using scoring. Currently only .Zip, .7z and .Rar files container are used, however this can be extended to .Arj, .Cab and .Tar files. Excluding/Whitelisting If you wish to whitelist one of the above signatures, you can do this by creating your own foxhole.ign2 file and place it in the ClamAV database folder: Example: printf "Sanesecurity.Foxhole.7z_avi" > foxhole.ign2 Restart clamd and the Sanesecurity.Foxhole.7z_avi signature will be ignored.	High
The following databases are distributed by Sanesecurity, but produced by OITC		
winnow_malware.hdb	Current virus, trojan and other malware not yet detected by ClamAV. Undetected virus samples can be sent to virus_samples@oitc.com	Low
winnow_malware_links.ndb	Links to malware	Low
winnow_spam_complete.ndb	Signatures to detect fraud and other malicious spam	Med
winnow_phish_complete.ndb	Phishing and other malicious url's and compromised hosts	High
winnow_phish_complete_url.ndb	Similar to winnow_phish_complete.ndb except that entire urls's are used	Med
winnow.complex.patterns.ldb	contain hand generated signatures for malware and some egregious fraud	Med
winnow_extended_malware.hdb	contain hand generated signatures for malware.	Low
winnow_extended_malware_links.ndb	contain hand generated signatures for malware links.	Med
winnow.attachments.hdb	Spammed attachments such as pdf's/docs/rtf/zips	Low

winnow_bad_cw.hdb	MD5 hashes of malware attachments acquired directly from a group of botnets	Low
Note: the two databases winnow_phish_complete.ndb and winnow_phish_complete_url.ndb shouldn't be used together.		
The following databases are distributed by Sanesecurity, but produced by Julian Field		
scamnailer.ndb	Spear phishing and other phishing emails	Med
The following databases are distributed by Sanesecurity, but produced by Doppelstern Antispam		
doppelstern.ndb	phishing, scams and other junk	Med
doppelstern.hdb	Hashes of spam documents and images	Low
The following databases are distributed by Sanesecurity, but produced by bofhland		
bofhland_cracked_URL.ndb	Spam URLs	Low
bofhland_malware_URL.ndb	Malware URLs	Low
bofhland_phishing_URL.ndb	Phishing URLs	Low
bofhland_malware_attach.hdb	Malware Hashes	Low
The following databases are distributed by Sanesecurity, but produced by CRDF		
crdfam.clamav.hdb	List of new threats detected by CRDF Anti Malware.	Low
The following databases are distributed by Sanesecurity, but produced by Porcupine Signatures		
porcupine.ndb	Brazilian e-mail phishing and malware signatures.	Low
phishtank.ndb	Online and valid phishing urls from phishtank.com data feed.	Low
The following databases are produced and distributed by SecuriteInfo		
honeynet.hdb	Old malwares not detected	Low
securiteinfoelf.hdb	Malwares ELF (Linux executables)	Low
securiteinfosh.hdb	Malwares SHELL (Linux)	Low
securiteinfopdf.hdb	Malwares PDF	Low
securiteinfooffice.hdb	Malwares Macros Office	Low
securiteinfohtml.hdb	Malwares HTML	Low
securiteinfodos.hdb	Malwares MS-DOS	Low
securiteinfobat.hdb	Malwares BAT	Low
securiteinfo.hdb	Malwares in the Wild	Low
spam_marketing.ndb	Spam Marketing	High
The following databases are produced and distributed by MalwarePatrol		
mbl.ndb	URLs containing of Viruses, Trojans, Worms, or Malware.	Low

This script is easily run via CRON. Something like this is all that you probably need. You should probably include a "MAILTO" in the crontab file. Any errors will be mailed to that address. If not all ready set, or if you do not know how to set it, at the command line enter: "whoami" sans quotes and enter that in the mailto variable.

EXAMBLE: Output of 'whoami' was steve. Place this in the top of the cron file:

MAILTO=steve

This would be placed just below the 'SHELL' variable.
You can get further information at <http://unixhelp.ed.ac.uk/CGI/man-cgi?crontab+5>

1. # Root's Crontab file
2. # Use the fully qualified path to bash on your system.
3. # Typing: "which bash" will produce it.
4. # SHELL=/usr/local/bin/bash # For FreeBSD users
5. # SHELL=/usr/bin/bash # Most other operating systems
6. # Enter user below and uncomment
7. MAILTO=
8. # (m) (h) (mday) (month) (wday) (command) ## Do NOT uncomment
9. # Runs every 1 hours, every day with logging=on, quiet mode=on, random download timer=on
   0 * * * * /PATH-2-SCRIPT/scamp.sh -L -q -R
10. # Runs every hour, every day using a preconfigured config file name: cron
    0 * * * * /PATH-2-SCRIPT/scamp.sh -C cron

EULA:

The end user is allowed to make any changes, modifications, or whatever to this script. The author assumes no responsibility for this script, modified or not by the end user. In other words, the user assumes all responsibility for the use of this program. In other words, ***USE AT YOU OWN RISK***.

I can be contacted at: gerard@seibercom.net

The latest version of this script can usually be downloaded from:

https://sourceforge.net/projects/scamp/

Older versions may also be available.

Any questions, suggestions, patches, etc. should be directed to me. I really would appreciate it. To make tracking of '**bug**' ☹ reports easier, please do the following:

1. Go to: https://sourceforge.net/projects/scamp/
2. Click on "TRACKER"
3. Click on "Bug Reports"
4. Click on "Add New"
5. Fill out the report with complete information including the version of the script you are using, your OS and version of **bash**, **rsync**, **gpg/gpg2**, **curl** and/or **wget** and your version of **Clamav**. If possible, include the complete text of any error messages, etc.

Use the same procedure to submit suggestions. Click on "Feature Request" under "TRACKER".

FreeBSD Note:

When installed via the *FreeBSD ports system*, a configuration file with the basic defaults for **Clamav** on a **FreeBSD** system is installed in the '/usr/local/etc/scamp' directory. It is still strongly recommended that the first time this script is run, it is run as **"scamp.sh -c"** to insure the file is configured according to the end users preferences.

COPYRIGHT

LAST UPDATED: Sat, 27 April 2013 19:41:43 GMT