avast! for linux technical documentation

Martin Tůma, tuma@avast.com

June 4, 2014

Contents

1 Overview

The avast! for linux products are a set of the following components. The components are distributed in the form of standard software packages - DEB for Debian/Ubuntu systems and RPM for RedHat/SUSE systems. Software repositories are also provided, so all the standard system management tools can be used to keep the avast programs up to date.

Packages

avast

The avast package provides the core scanner service (avast) and a command line scan utility (scan). It can be used for on demand scanning or mail server integration using AMaViS as described in section 6.

The avast package is required by the avast-proxy and avast-fss packages.

avast-proxy

The avast-proxy package provides a transparent network traffic filtering proxy designed for gateway/router usage. Using avast-proxy, you can scan all computer network traffic from a single machine. avast-proxy supports the HTTP, IMAP and POP3 protocols as well as their secured variants (HTTPS, IMAPS, POP3S) using certificate resigning.

Network traffic redirection is required for the proxy to work. This is done using iptables, the standard linux firewall (netfilter) interface. See the attached avast-proxy manual page for example iptables rules.

avast-fss

The avast-fss package provides a fanotify based "on write" filesystem shield designed for fileserver usage. The typical target field for avast-fss are SMB/NFS file servers.

Business products

The avast components are available as the following business products:

avast! Core Security

Is a license for the bare 'avast' package.

avast! File Server Security

Is a license for the 'avast' and the 'avast-fss' packages.

avast! Network Security

Is a license for the 'avast' and the 'avast-proxy' packages.

avast! Security Suite

Is a license for all three packages (avast, avast-fss, avast-proxy).

2 Installation

The avast! linux server products installation consist of two steps:

1. Add the avast repository to the system repositories.
2. Get the desired packages from the repository.

Debian/Ubuntu

1. Add the avast repository to the system repositories

```
# echo "deb http://deb.avast.com/lin/repo debian release" \
  >> /etc/apt/sources.list
# apt-key add /path/to/avast.gpg
# apt-get update
```

2. Install the avast package and optionally the avast-fss and avast-proxy packages.

```
# apt-get install avast
# apt-get install avast-fss
# apt-get install avast-proxy
```

RHEL/CentOS

1. Add the avast repository to the system repositories:

```
# echo '[avast]
name=Avast
baseurl=http://rpm.avast.com/lin/repo/dists/rhel/release
enabled=1
gpgcheck=1' > /etc/yum.repos.d/avast.repo
# rpm --import /path/to/avast.gpg
```

2. Install the avast package and optionally the avast-proxy package.

```
# yum install avast
# yum install avast-proxy
```

SUSE

1. Add the avast repository to the system repositories:

```
# echo '[avast]
name=Avast
baseurl=http://rpm.avast.com/lin/repo/dists/suse/release
enabled=1
gpgcheck=1' > /etc/yum.repos.d/avast.repo
# rpm --import /path/to/avast.gpg
```

2. Install the avast package

```
# yum install avast
# yum install avast-proxy
```

The avast GPG public key referenced as "avast.gpg" can be found in appendix E.

3 Operation

All avast packages provide conventional init scripts for starting/stopping the services. For example starting the core avast service is done by running

```
# /etc/init.d/avast start
```

and stopping the service is done by running

```
# /etc/init.d/avast stop
```

All avast services use the system logger (syslog) for logging, so the log file location is dependent on the host system. The most common log file paths are /var/log/messages and /var/log/syslog.

4 Licensing

The access to the program repositories is not restricted in any way, you can always obtain all the latest packages, but for running the components, a license file is required. The license for the products comes in the form of a file named "license.avastlic". After obtaining the license file, copy it into the /etc/avast directory:

```
# cp /path/to/license.avastlic /etc/avast
```

5 Virus definitions updates

Regularly updating the virus definitions database (VPS) is necessary to keep your antivirus protection up to date. Avast antivirus provides a shell script, which checks for the latest VPS over the Internet and takes care of its downloading and installing. The update script is automatically installed and periodically executed every hour as a cronjob.

The default avast crontab entry is:

```
0 * * * * /var/lib/avast/Setup/avast.vpsupdate
```

Avast uses incremental updates, so the average update data size is less than 0.5MB.

Local virus definitions mirrors

It is possible to use a local, mirrored, VPS repository. This is useful, if you are runing several avast instalations in your local network.

To set up a local VPS mirror, all you need is a local HTTP server that can serve a copy of the official public repository. To get your local repository copy, use the following command:

```
$ wget -m "http://download.ff.avast.com/lin/‘uname -i‘"
```

To change the VPS repository URL, that avast uses for getting the VPS updates edit the /etc/avast/vps.conf configuration file.

6 AMaViS integration

AMaViS is an interface between mailer (MTA) and content checkers, which is already prepared for integration with mail scanners. This section describes how to integrate avast into AMaViS.

Integration of avast into AMaViS covers updating AMaViS configuration and enabling access to emails going through AMaViS to be scanned by avast. This can be divided into these three steps:

1. Integrating avast antivirus

   Open the AMaViS configuration file (e.g. /etc/amavis/conf.d/50-user) and insert the following six lines into the file:

   ```
   @av_scanners = (
     ### http://www.avast.com/
     ['AVAST Antivirus - Client/Server Version', '/bin/scan',
       '{}', [0], [1],
       qr/\t(.+)/m ]
   );
   ```

2. Enabling virus scanning

   Then open the AMaViS content filter configuration file (e.g. /etc/amavis/conf.d/15-content_filter_mode) and enable antivirus checking mode by uncommenting the 'bypass_virus_checks' lines.

3. Update of access permissions

   Finally enable the avast daemon to scan emails going through AMaViS:

   ```
   # usermod -G amavis -a avast
   ```

Appendices

A scan manual page

```
       scan - avast! antivirus command line scan utility

SYNOPSIS
       scan [-s SOCKET] [-apf] [-e PATH] PATH...
       scan [-s SOCKET] -V
       scan -h | -v

DESCRIPTION
       Scan  is  the  basic command line scanner that comes with
       avast! antivirus for Linux/OS X. It  searches  the  given
       PATH(s)  for  infected  files and eventually reports such
       files to the standard output.

OPTIONS
       -h     Print short usage info and exit.

       -v     Print program version and exit.

       -V     Print the  virus  definitions  (VPS)  version  and
              exit.  The VPS version is retrieved from the avast
              service.

       -s SOCKET
              Use SOCKET to connect to the avast  scan  service.
              The     default     scan     socket     path    is
              /var/run/avast/scan.sock.

       -a     Print all scanned files, not only infected.

       -p     Print archive content. If set, the files in an ar-
              chive  are  listed  separately,  each with its own
              scan status.

       -b     Report decompression bombs as infections. If  set,
              files  suspected  of being decompression bombs are
              reported as infected, not as errors.

       -u     Report potentionally unwanted programs  (PUP).  If
              set, PUP files are reported as infected.

       -e PATH
              Exclude PATH from the scan. Use this option multi-
              ple  times,  if  more  than  one  exclude  path  is
              required.

       -f     Scan full files. If set, the whole file content is
              scanned, not only the relevant file parts.
```

OUTPUT FORMAT

Every detected malicious file is reported on a separate line in the format:

PATH INFECTION

where PATH and INFECTION are separated by a TAB character. If all files are printed using the -a option, then the clean files have a "[OK]" string as the infection name and files that could not be scanned (insufficient permissions, corrupted archives, ...) have an "[ERROR]" string as the infection name.

If the -p option is set, PATH contains the archive path delimited by a "|>" delimiter in case of an archive.

EXIT STATUS

The exit status is 0 if no infected files are found and 1 otherwise. If an error occurred, the exit status is 2.

SEE ALSO

avast(1)

B avast manual page

```
       avast - avast! antivirus scanner

SYNOPSIS
       avast [OPTIONS]

DESCRIPTION
       avast  is  an  antivirus scan service for OS X and Linux.
       Clients (shields, command line scan tool, ...) connect to
       the  services's UNIX socket and perform scan requests and
       receive scan results.

OPTIONS
       -h     Print short usage info and exit.

       -v     Print the program version and exit.

       -c FILE
              Set configuration file path to FILE.  The  default
              configuration file is /etc/avast/avast.conf.

       -n     Do not daemonize.

CONFIGURATION
       The configuration file format is INI file format, i.e. it
       consists of KEYWORD = VALUE entries, each on  a  separate
       line. Lines begining with ';' are treated as comments and
       are ignored. Keys may be grouped into  arbitrarily  named
       sections.  The  section name appears on a line by itself,
       in square brackets ([ and ]).

       The following example is an avast configuration file with
       explicitly defined default options:

       ; avast! configuration file

       RUN_DIR = "/var/run/avast"
       TEMP_DIR = "/tmp"
       DATA_DIR = "/var/lib/avast"
       SOCKET = "/var/run/avast/scan.sock"
       LICENSE = "/etc/avast/license.avastlic"
       SUBMIT = "/var/lib/avast/Setup/submit"

       [OPTIONS]
       CREDENTIALS = 0
       STATISTICS = 1
       HEURISTICS = 1
       STREAMING_UPDATES = 1

       The  configuration  file  is re-read on HUP signal by the
       program, but only the entries in the Options section  are
```

reloaded, changes to the global parameters are ignored.

Global parameters

RUN_DIR
: Run directory. The PID file is stored here.

TEMP_DIR
: Temporary directory. The program temporary files are stored here.

DATA_DIR
: Data directory. Contains the virus definitions database and various other data files used by avast.

SOCKET Path to the UNIX socket used by the clients to connect to the scan service. The socket is created by avast at service start.

LICENSE
: Path to the license file.

SUBMIT Path to the submit utility. If enabled (see the Options section), the submit utility creates and sends reports about infected and suspicious files to the avast virus lab.

Options

CREDENTIALS
: If enabled, avast performs a UNIX socket credentials check, whenever a new client is connecting. If the client's effective UID does not match the effective UID of the avast process or the client's effective GID does not match the avast effective GID or any avast suplementary group GID, the connection is refused.

STATISTICS
: If enabled, avast creates statistics submits about detected malicious files.

HEURISTICS
: If enabled, avast creates heuristics submits about suspicious files detected during the scan.

STREAMING_UPDATES
: If enabled, the scan service establishes a permanent network connection to the avast cloud and retrievs virus definitions updates instantly as they are released. Streaming updates are an addition to the regular virus database updates, they do not replace them (you always get all the

streamed updates in the next regular virus definitions database update).

SEE ALSO

scan(1)

C avast-fss manual page

```
       avast-fss - avast! file server shield

SYNOPSIS
       avast-fss [OPTIONS]

DESCRIPTION
       avast-fss,  a  part  of avast! antivirus for Linux suite,
       provides real-time scanning of files written  to  any  of
       the monitored mountpoints. avast-fss is based on the fan-
       otify access notification system available on Linux  ker-
       nels 2.6.37+.

OPTIONS
       -h     Print short usage info and exit.

       -v     Print the program version and exit.

       -c FILE
              Set  configuration file path to FILE.  The default
              configuration file is /etc/avast/fss.conf.

       -n     Do not daemonize.

CONFIGURATION
       The configuration file  format  is  INI  file  format  as
       described in the avast(1) manual page.

       The configuration consists of two parts - the global con-
       figuration options and the monitoring configuration.  The
       sample  configuration  below  shows  all available global
       configuration options and their default  values  followed
       by  some  examples of monitoring (and monitoring exclude)
       entries.

       ; avast! fileserver shield configuration file

       RUN_DIR = "/var/run/avast"
       SOCKET = "/var/run/avast/scan.sock"
       LOG_FILE = "/var/log/avast/fss.log"
       CHEST = "/var/lib/avast/chest"
       SCANNERS = 4

       [MONITORS]
       SCAN = "/some/mountpoint/to/monitor"
       SCAN = "/another/mountpoint/to/monitor"
       EXCL = "/path/to/exclude/from/scan"

   Global parameters
       RUN_DIR
              Run directory. The PID file is stored here.
```

SOCKET Path to the avast service UNIX socket.

LOG_FILE
Path to the virus log file.

CHEST Path to the chest directory. The chest directory is where the detected malicious files are moved. If the chest directory is located on a monitored mountpoint, it is automatically added to the excluded paths on startup.

SCANNERS
Number of parallel running scans. Set this option to the number of CPU cores to get the best performance.

Monitors
SCAN A mountpoint (path) that shall be monitored by avast-fss. If the given path is not a system mountpoint, it is automatically converted to the corresponding mountpoint.

EXCL A path to be excluded from monitoring.

SEE ALSO
avast(1), fanotify(7)

D avast-proxy manual page

avast-proxy - avast! network shield

SYNOPSIS

avast-proxy [OPTIONS]

DESCRIPTION

avast-proxy, a part of the avast! antivirus for Linux suite, provides real-time network traffic scanning. The network shield is technicaly a transparent proxy that filters the traffic that goes through it. The system firewall (iptables) is used to redirect the network traffic so it goes through the proxy.

Secured connections

The proxy is capable of scanning secured connections (https, imaps, pop3s), if enabled in the configuration.

During the installation, 2 SSL CA certificates are generated. One is called avast! trusted CA and the other one called avast! untrusted CA. The avast! trusted CA certificate must be distributed to the clients that are using the proxy and put there into the system keychain and/or browser SSL certificate storage. Unlike the trusted CA certificate, the avast! untrusted CA certificate MUST NOT be exported to the clients!

On a secured connection, the proxy does the initial SSL handshake with the destination server, checks the server's SSL certificate and sends a "recreated" certificate signed by either the avast! trusted CA or avast! untrusted CA to the client. Verified certificates are resigned with the avast! trusted CA certificate. Certificates where the issuer certificate of a locally looked up certificate could not be found or self signed certificates are resigned with the avast! untrusted CA certificate. Expired, revoked, or not valid at all certificates are not resigned and the connection is dropped.

OPTIONS

-h Print short usage info and exit.

-v Print the program version and exit.

-c FILE

Set configuration file path to FILE. The default configuration file is /etc/avast/proxy.conf.

-n Do not daemonize.

CONFIGURATION

The configuration file format is INI file format as described in the avast(1) manual page.

The configuration consists of two parts - the global configuration options and the module (protocol) configurations. Available modules are http, https, pop3, pop3s, imap and imaps. For a sample configuration file, see the EXAMPLE section. For default values, see the suplied proxy.conf configuration file in /etc/avast.

Global parameters

RUN_DIR
: Run directory. The PID file is stored here.

TEMP_DIR
: Temporary directory. The program temporary files are stored here.

DATA_DIR
: Resources directory. The HTML page templates are stored here.

CERT_DIR
: A directory of trusted certificates. Equivalent to the -CApath option of OpenSSL verify(8).

CERT_FILE
: A file of trusted certificates. The file should contain multiple certificates in PEM format concatenated together. Equivalent to the -CAfile option of OpenSSL verify(8).

CA_DIR Proxy CA storage directory. The issued resigned certificates are stored here.

CA_CERT
: avast! trusted CA certificate file.

CA_KEY avast! trusted CA private key file.

UCA_CERT
: avast! untrusted CA certificate file.

UCA_KEY
: avast! untrusted CA private key file.

EXCEPT_FILE
: Exceptions file. Contains a list of addresses that shall not be scanned by the corresponding service handler. The exceptions file is a simple text file with one entry per line in the format: HOSTNAME

SERVICE. Fields of the entry are separated by any number of blanks and/or tab characters. Text from a "#" character until the end of the line is a comment, and is ignored. For details about HOSTNAME and SERVICE notation see getaddrinfo(3).

Note, that the proxy works on IP level so although HOSTNAME can be given as a domain name, the proxy always compares IP addresses (translated on program startup), when checking for exceptions. This is why using exceptions on hosts using dynamic DNS will not work.

SOCKET Path to the avast service UNIX socket.

LOGFILE
Path to the virus log file.

SCANNERS
Maximal number of parallel running scans.

HANDLERS
Maximal number of simultaneous connections.

ADDRESS
IPv4 listen address. The default IPv4 listen address is 0.0.0.0.

ADDRESS6
IPv6 listen address. The default IPv6 listen address is ::0.

Modules
ENABLED
Enable/disable the module. All available modules are enabled by default.

IPV6 Enable/disable IPv6. Note: IPv6 support requires linux kernel >= 3.8 and iptables >= 1.4.17. IPv6 support is disabled by default.

PORT Module listen port. The default value is the service port (e.g. 80 for HTTP) plus 12000, i.e. 12080 for the http module.

LIMIT Do not scan files > LIMIT bytes, 0 = no limit.

EXAMPLE
The sample configuration below shows a typical gateway setup configuration that filters HTTP/HTTPS traffic on both IPv4 and IPv6.

```
; avast! network shield configuration file

HANDLERS = 256
SCANNERS = 32

[http]
IPV6 = 1
LIMIT = 67108864
[https]
IPV6 = 1
LIMIT = 67108864

[pop3]
ENABLED = 0
[pop3s]
ENABLED = 0
[imap]
ENABLED = 0
[imaps]
ENABLED = 0
```

The apropriate firewall setup for a system with eth0 as the internal zone (the network where we want to check the traffic):

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
-j REDIRECT --to-ports 12080
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 \
-j REDIRECT --to-ports 12443
ip6tables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
-j REDIRECT --to-ports 12080
ip6tables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 \
-j REDIRECT --to-ports 12443
```

SEE ALSO

avast(1), iptables(8), getaddrinfo(3), verify(1)

E avast GPG public key

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.12 (GNU/Linux)

mQENBFMoeIMBCACnCOmAfky/Mla7p2VpDrPtCWdjsMQm+Frf9fVRcgNvZYzexrGv
Qun7tDgCELyAYmElYg/45YeqRT+l5fxpVwG0Unz7jYnHWxt16ojZL2eKI85QDkox
2UUdEkYq8ruECirpg+IUenrO0UQpZKqgx+IYgYQfWrh0cbrKziO0/GCEGpwnI0Iu
lh283mD/AvxY3DyvBjNFk1en1zFFJV5Df4ppZF1vWkIVbv23VDXyooLYNSXk1yJ/
zXLF50p3ex4tdlkGV6ce64iShlO2yfp/36vCyBVsCL8Y4dEeSQZu+4bPkVmyUV75
QmtlbOEDOqdC8MEImGd/s2uoJP1HFl1SUKKvABEBAAG0OkF2YXN0IFNvZnR3YXJl
IHMuci5vLiAoUmVsZWFzZSBFbmdpbmVlcmluZykgPHJlQGF2YXN0LmNvbT6JATgE
EwECACIFAlMoeIMCGwMGCwkIBwMCBhUIAgkKCwQWAgMBAh4BAheAAAoJEJHuE/BX
Ty17OhAIAKA/vGSTWvT1Bm049fwNudWXxBc3l97meqa0DVTv2TzCOiK3W5w/CKUQ
RaTXYHpak6lbRMeRu8kShvlKBJ15CsoKUSz0zTgrwxmDhiYBcsafh0R81+51jElI
YxAZfBkKZtI4RJxXFpbOVVe9AeOnMgTdFreNK/E0tjZQStNUKl1W7kDPV3W3eVbY
JAdUbBHBvqvkBHZ90B0kgeOofHZ2z1GQCCc1ClxSw2n0WDFlQ96cfSL8YeHb1bbF
P+hMW1V1L6lgN7VdpfhsOdGhzPb9VCU4K/pGzSSNeg1ksVCH2bm+7Y8AoX2BSVDT
5UbYbrt9AdDES9nuKSmFrqgbtdxZJO65AQ0EUyh4gwEIANI4a5lOnaA/mRySIIIm
JMovZJzH5muO0ao7D3SiWtyT7DSPo6LzO3eLnC0AjZ+dT14kVKiekRNMD/3cSPNP
2ulbeTe00RbmaCz30w+vWWdt2IWKGB8whvkUh/4dzbY49FHek0+WkaLJRD1UIUE5
13ICmU6m7xeMv64tN3cWwuEYjQoJLRQezR1u0GU+0MSDv3J813WwZbxU5XYX71hO
2G/CD9utu4eUl0MpPBv5x9e1sPjUET6e0xS7RmRzk4mxBAiUtIT2RcOELghPj1q7
oNBuaUkeHhx5aebokJKxzekto8fpjRo7OGlVe/QlZxL1UD+QxyVPfVNpVyOUHvYI
qzsAEQEAAYkBHwQYAQIACQUCUyh4gwIbDAAKCRCR7hPwV08te0RCB/9vF538ooRD
bgRBBN5mviKxuxFnrEQYsPpZvmEsHvS6RSQfPvmVF3z4HUoKHWFsqRbhaJCRVWbm
fl8X8DOezAVR734MYaicj+NzVdKAKWu+a5TJ5XxVG2mSY+a0PK3FkF4cSH2fgmxq
q/NiYFVY2SZpwEOg+zkYF8m1+DoxSpeJ7wapPcFhgIt5YS6Bego6AM1Ork2yTXY7
95ZMFyFjT9XJJUo9BG4NMnzVxsgMhJ6g1zGKtsoVrPgxmyJ5KHA+Hr5BvkESuXQw
mQp5EeiKUqxAWe7wbk59oSKUNYAJen/X3jCCYaXqN1vEX5E6kcZO2O6e2II32ecP
r4XP+TMQpz3L
=nuBs
-----END PGP PUBLIC KEY BLOCK-----
```