



# avast! Security for Linux

TECHNICAL DOCUMENTATION

# CONTENTS

1. Overview	01
2. Installation	02
3. Operation	03
4. Licensing	03
5. Virus definitions updates	03
6. AMaViS integration	04
Appendices	05
A) scan manual page	05
B) avast manual page	06
C) avast-fss manual page	07
D) avast-proxy manual page	08
E) avast GPG public key	11

# 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 4. 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 fileservers 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
```

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 running several avast installations 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

**NAME** 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 archive 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.  
**-e PATH** Exclude PATH from the scan. Use this option multiple 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 the 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

**NAME** 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 beginning 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
```

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 supplementary 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.

### SEE ALSO

scan(1)

## C) AVAST-FSS MANUAL PAGE

**NAME** 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 fanotify access notification system available on Linux kernels 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 configuration 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 mount point, 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 mount point (path) that shall be monitored by avast-fss. If the given path is not a system mount point, it is automatically converted to the corresponding mount point.

**EXCL** A path to be excluded from monitoring.

### SEE ALSO

avast(1), fanotify(7)

## D) AVAST-PROXY MANUAL PAGE

**NAME** 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 technically 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 is called "avast! untrusted CA". The avast! trusted CA certificate must be distributed to the clients 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 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 supplied 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.
- 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.

<b>EXCEPT_FILE</b>	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 <code>getaddrinfo(3)</code> .
<b>SOCKET</b>	Path to the avast service UNIX socket.
<b>LOGFILE</b>	Path to the virus log file.
<b>SCANNERS</b>	Maximal number of parallel running scans.
<b>HANDLERS</b>	Maximal number of simultaneous connections.
<b>ADDRESS</b>	IPv4 listen address. The default IPv4 listen address is 0.0.0.0.
<b>ADDRESS6</b>	IPv6 listen address. The default IPv6 listen address is ::0.

### Modules

<b>ENABLED</b>	Enable/disable the module. All available modules are enabled by default.
<b>IPV6</b>	Enable/disable IPv6. Note: IPv6 support requires linux kernel $\geq 3.8$ and iptables $\geq 1.4.17$ . IPv6 support is disabled by default.
<b>PORT</b>	Module listen port. The default value is the service port (e.g. 80 for HTTP) plus 12000, i.e. 12080 for the http module.
<b>LIMIT</b>	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 appropriate 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+Fr9fVVRcgNvZYzextrGv  
Qun7tdgCELyAYmElYg/45YeqRT+l5fxpVwG0Unz7jYnHWxt16ojZL2eKI85QDkox  
2UudEkYq8ruECirpg+IUenr00UQpZKqgx+IYgYQfWrh0cbrKzi00/GCEGpwnI0Iu  
lh283mD/AvxY3DyvBjNFk1en1zFFJV5Df4ppZF1vWkIVbv23VDXyooLYNSXk1yJ/  
zXLF50p3ex4tdlkGV6ce64iShl02yfp/36vCyBVCL8Y4dEeSQZu+4bPkVmyUV75  
Qmt1b0EDOqdC8MEImGd/s2uoJP1HF11SUKKvABEBAAG0kF2YXN0IFNvZnR3YXJl  
IHMuci5vLiAoUmVsZWZzZSBfbmdpbmVlcm1uZykgPHJlQGF2YXN0LmNvbT6JATgE  
EwECACIFAlMoeIMCGwMGcWkIBwMCBhUIAgkKCwQWAgMBAh4BAheAAoJEJHuE/BX  
Ty170hAIAKA/vGSTwvT1Bm049fwNudWxXbc3l97meqa0DVTv2TzC0iK3W5w/CKUQ  
RaTYHpak6lbRMeRu8kShv1KBJ15CsoKUSz0zTgrwxmDhiYBcsafh0R81+51jElI  
YxAZfBkKztI4RjXfPb0VVe9AeOnMgTdfrenK/E0tjZQStNUK11W7kDPV3W3eVbY  
JAdUbBHBvqvkBHZ90B0ke0ofHZ2z1GQCCc1ClxSw2n0WDF1Q96cfSL8YeHb1bbF  
P+hMW1V1L6lgN7Vdpfhs0dGhzPb9VCU4K/pGzSSNeg1ksVCH2bm+7Y8AoX2BSVDT  
5UbYbrt9AdDES9nuKSmFrqgbtdxZJ065AQ0EUyh4gwEIANI4a5l0naA/mRySIIIm  
JMovZJzH5mu00ao7D3SiWtyT7DSPo6Lz03eLnC0AjZ+dT14kVKiekRNMD/3cSPNP  
2ulbeTe00RbmaCz30w+vwWdt2IWKGB8whvkUh/4dzbY49Fhek0+WkaLJRD1UIUE5  
13ICmU6m7xeMv64tN3cWwuEYjQoJLRQezR1u0GU+0MSDv3J813WwZbxU5XYX71h0  
2G/CD9utu4eU10MpPBv5x9e1sPjUET6e0xS7RmRzk4mxBAiUtIT2Rc0ELghPj1q7  
oNBuaUkeHx5aebokJKxzekto8fpjRo70G1Ve/Q1ZxL1UD+QxyVPfVnpVy0UHvYI  
qzsAEQEAAyKBHwYQAQIACQUcUyh4gwIbDAAKCR7hPwV08te0RCB/9vF538ooRD  
bgRBBN5mviKxuxFnrEQYsPpZvmEsHvS6RSQfPvmVF3z4HUoKHWfsqRbhaJCRVWbm  
fl8X8D0ezAVR734MYaicj+NzVdKAKWu+a5TJ5XxVG2mSY+a0PK3FkF4cSH2fgmxq  
q/NiYFVY2SZpwE0g+zkyF8m1+DoxSpeJ7wapPcFhgIt5YS6Bego6AM10rk2yTXY7  
95ZMFyFjT9XJJUo9BG4NMnzVxsgMhJ6g1zGktsoVrPgxyJ5KHA+Hr5BvkESuXQw  
mQp5EeiKUqxAWe7wbk59oSKUNYAjen/X3jCCYaXqN1vEX5E6kcZ0206e2II32ecP  
r4XP+TMQpz3L
```

=nuBs

-----END PGP PUBLIC KEY BLOCK-----

## NEED HELP?

If you have any questions, ideas or suggestions, please do not hesitate to contact us on [reseller@avast.com](mailto:reseller@avast.com).

Thank you for your interest in avast! solutions.

Best regards,  
The avast! team



avast software a.s.  
trianon office building  
budějovická 1518/13a  
140 00 praha 4  
czech republic

**TELEPHONE** +420 274 005 666  
**FAX** +420 274 005 889  
**WEB** [www.avast.com](http://www.avast.com)  
**TWITTER** [twitter.com/avast\\_antivirus](https://twitter.com/avast_antivirus)  
**FACEBOOK** [facebook.com/avast](https://facebook.com/avast)

avast! is a registered trademark of AVAST Software s.r.o. All other brands and products are trademarks of their respective holder(s).  
Copyright © 2014 AVAST Software s.r.o. All product information is subject to change without notice.