
Armadito Antivirus Documentation

Release 1.0

Teclib

February 27, 2017

1	Introduction	3
2	Compilation	5
2.1	Compilation on Linux	5
2.2	Compilation on Windows	10
3	Installation	15
3.1	Installation on Linux	15
3.2	Installation on Windows	15
4	Configuration	17
4.1	Configuration on Linux	17
4.2	Configuration on Windows	19
5	Contribute	21
6	Licensing	23
7	FAQ	25
8	Screenshots	27
8.1	Status board	29
8.2	Scan	29
8.3	Journal	29



Armadito Antivirus is an open-source antivirus that protects your computers and servers from malware and viruses. It includes classical signature-based malware detection and provides innovative heuristic detection modules for binaries (MS-Windows and GNU/Linux) and for PDF documents.

Introduction

Armadito Antivirus is an open-source antivirus that protects your computers and servers from malware. It includes classical signature-based malware detection and provides innovative heuristic detection modules for binaries (MS-Windows and GNU/Linux) and for PDF documents.

This project is designed in a way which allows anyone to improve anti-malware techniques' state of the art. For example, security experts are able to concentrate on anti-malware techniques only by developing their own modules in C language. This plugin system aims to encourage people to test and develop experimental techniques freely.

An intuitive and user-friendly interface gives access to all Armadito's features: on-demand scanning, real-time protection, quarantine zone, threat detection journal, etc. This interface has been developed using AngularJs/HTML5/CSS3 technology. Consequently, this graphical user interface aims to be multi-platform (Windows, Linux, Mac OS X).

Compilation

Compilation on Linux

On linux, Armadito AV's sources are compiled using automake and autoconf to generate Makefiles. You can compile each part separately by ourself.

Sources are publicly available on github.com, in separate git repositories :

- Core : <https://github.com/armadito/armadito-av>
- WebUI : <https://github.com/armadito/armadito-web-ui>
- SystrayUI : <https://github.com/armadito/armadito-systray-ui>
- Module Clamav : <https://github.com/armadito/armadito-mod-clamav>
- Module PDF : <https://github.com/armadito/armadito-mod-pdf>
- Module H1 : <https://github.com/armadito/armadito-mod-h1>
- Module Yara : <https://github.com/armadito/armadito-mod-yara>

Armadito core

Armadito core corresponds to libarmadito library. Symbols exported from this library allows all modules libraries to use the same APIs.

Sources are publicly available on github.com, you can get it with the following command :

```
$ git clone https://github.com/armadito/armadito-av.git -b DEV
```

Prerequisites

In order to build libarmadito, you need the following tools:

- automake/autoconf
- GNU make
- C compiler
- glib, libmagic, libxml2, libmicrohttpd, libcurl libraries and headers

To install the needed headers:

- Ubuntu:

```
apt-get install automake autoconf libtool libglib2.0-dev libmagic-dev libxml2-dev libjson-c-dev libmicrohttpd-dev
```

- Fedora:

```
dnf install automake autoconf libtool glib2-devel file-devel libxml2-devel json-c-devel libmicrohttpd-devel
```

Configuration

Once git repo cloned, you need to initialize the build using automake, autoconf and tools. A shell script **autogen.sh** is provided to ease this step:

```
$ ./autogen.sh
+ aclocal --force
+ libtoolize --force --automake --copy
+ autoheader --force
+ automake --foreign --add-missing --force-missing --copy
+ autoconf --force
```

This will generate the **Makefile.in** files and the **configure** script.

configure script takes the following useful options:

- prefix=PREFIX** install architecture-independent files in PREFIX [default is /usr/local]
- enable-debug** enable debugging [default is yes]

The **PREFIX** directory will be used by **make install**. Its use is mandatory, unless building a package and installing in system directories, since building the scanning modules and the graphical user interface will need a libarmadito properly installed.

Building in a separate directory is highly recommended, unless you really want to clobber the source tree with objects, libraries, binaries and other stuff.

```
$ mkdir /home/joebar/build/libarmadito
```

Typical invocation of the configure script is:

```
$ /home/joebar/armadito-av/libarmadito/configure --prefix=/home/joebar/install --enable-debug
```

Building

Once configured, build is easy:

```
$ make
```

Installing

After build, installation is done by:

```
$ make install
```

This will install libraries, tools, header files... in the subdirectories of the **PREFIX** directory defined at configure time.

Armadito modules

Building is quite the same for each Armadito Antivirus' analysis modules. Build/Install all modules is not mandatory. Everyone has the choice of which modules to use within Armadito Antivirus.

Warning: Be sure you have built **Armadito core** library before trying to build any of these modules.

Prerequisites

In order to compile Armadito ClamAV module, you need the following tools:

- automake/autoconf
- GNU make
- C compiler
- libarmadito (core)

Some modules have more external dependencies :

armadito-mod-clamav :

- libclamav library and headers. Ubuntu : libclamav-dev

armadito-mod-yara :

- libyara library and headers. Ubuntu : libyara-dev

armadito-mod-h1 :

- pas de dépendances supplémentaires

armadito-mod-pdf :

- pas de dépendances supplémentaires

Clone

```
git clone -b DEV https://github.com/armadito/armadito-mod-clamav.git
git clone -b DEV https://github.com/armadito/armadito-mod-yara.git
git clone -b DEV https://github.com/armadito/armadito-mod-h1.git
git clone -b DEV https://github.com/armadito/armadito-mod-pdf.git
```

Configuration

To initialize the build using automake, autoconf and tools, a shell script **autogen.sh** is provided to ease this step :

```
$ ./autogen.sh
+ aclocal --force
+ libtoolize --force --automake --copy
+ automake --foreign --add-missing --force-missing --copy
+ autoconf --force
```

This will generate the **Makefile.in** files and the **configure** script.

configure script takes the following useful options:

--prefix=PREFIX install architecture-independent files in PREFIX [/usr/local]

--enable-debug enable debugging [default is yes]

The **PREFIX** directory will be used by **make install**. Its use is mandatory, unless building a package and installing in system directories.

libarmadito use the **pkg-config** utility to specify compiling options relative to libarmadito. Since the **libarmadito.pc** specification file for **pkg-config** is not located in standard directory (usual **/usr/lib/pkgconfig**), invoking the configure script must use the **PKG_CONFIG_PATH** environment variable.

Building in a separate directory is highly recommended, unless you really want to clobber the source tree with objects, libraries, binaries and other stuff.

```
$ mkdir /home/joebar/build/modules/clamav
```

Typical invocation of the configure script is:

```
$ REPO_GIT/configure --prefix=/home/joebar/install --enable-debug PKG_CONFIG_PATH=/home/joebar/install
```

Note that the path specified in the value of **PKG_CONFIG_PATH** must be coherent with the **PREFIX** used in libarmadito installation (see **Armadito core** linux build section).

Building

Once configured, build is easy:

```
$ make
```

Installing

After build, installation is done by:

```
$ make install
```

This will install libraries, tools, header files... in the subdirectories of the **PREFIX** directory defined at configure time.

Armadito WebUI

Armadito WebUI relies on recent web technologies to provide a multi-plateform graphical user interface for Armadito antivirus.

Sources are publicly available on github.com, you can get it with the following command :

```
$ git clone https://github.com/armadito/armadito-web-ui.git -b DEV
```

Prerequisites

To run Armadito graphical user interface, you need:

- bower

Installing node and bower

Installations are done as root.

Installing bower (must be done as root too):

```
$ npm install -g bower
```

Installing modules in source tree

After cloning the repository, the source tree of the user interface must be configured :

```
$ cd /home/joebar/armadito-web-ui
$ bower install
```

Configuration

Once git repo cloned, you need to initialize the build using automake, autoconf and tools. A shell script **autogen.sh** is provided to ease this step:

```
$ ./autogen.sh
+ aclocal --force
+ automake --foreign --add-missing --force-missing --copy
+ autoconf --force
```

This will generate the **Makefile.in** files and the **configure** script.

configure script takes the following useful options:

--prefix=PREFIX install architecture-independent files in PREFIX [default is /usr/local]

The **PREFIX** directory will be used by **make install**. Its use is mandatory, unless building a package and installing in system directories, since building the scanning modules and the graphical user interface will need a libarmadito properly installed.

Typical invocation of the configure script is:

```
$ /home/joebar/armadito-web-ui/configure --prefix=/home/joebar/install
```

Building

Once configured, build is easy:

```
$ make
```

Installing

After build, installation is done by:

```
$ make install
```

This will install libraries, tools, header files... in the subdirectories of the **PREFIX** directory defined at configure time.

Running the interface

First, the Armadito daemon must be launched.

Open your web browser and go to the following URL :

<http://localhost:8888/app/index.html>

Debugging the interface

Once the interface is launched:

- right-click in the window to display debug menu and select “Inspect” or tap F12
- in the inspector window, select the “console” tab

Build with grunt

Install grunt :

```
npm install -g grunt-cli
```

Run *grunt* for building and *grunt serve* for preview.

You can use “-force” if you want to build with warnings.

Compilation on Windows

Danger: Graphical user interface is under redesign and new version is not yet ported to Windows. That will be fixed as soon as possible.

On Windows, you can compile Armadito AV sources with Visual Studio. This has been tested with Visual Studio 2013. You might have to apply some modifications regarding to which Visual Studio’s version you use.

Tested on : Windows 7 64 bit with Service Pack 1.

Armadito solution for Visual Studio is divided in the following subprojects :

- **Driver**
 - ArmaditoGuard (driver sources)
 - ArmaditoGuard Package (driver package)
- **Libarmadito**
 - libarmadito (armadito core library)
- **Modules**
 - clamav_a6o (clamav module)
 - moduleH1 (heuristic module)
 - modulePDF (PDF module)
- **Service**
 - ArmaditoSvc (analysis service)
- **Setup**
 - ArmaditoGuard-setup (driver installation)
 - Armadito-db-setup (setup project for module databases)
 - Armadito-setup (setup project for armadito)

Armadito core

Armadito core corresponds to libarmadito library. Symbols exported from this library allows all modules to use same API. On Windows, after build, a library called **libarmadito.dll** will be generated. For more simplicity, dependencies have been regrouped into a single zip archive automatically generated.

Prerequisites

- Microsoft Visual Studio 2013 (Community edition or more)
- Armadito windows dependencies archive (deps-x.zip)

Uncompress **deps-x.zip** in armadito-av sources root directory. You should have then these exact dependencies paths :

```
SOMEWHERE\armadito-av\deps\glib\...
SOMEWHERE\armadito-av\deps\json-c\...
```

Build

Open the armadito-av VS solution at location :

```
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\Armadito-AV.sln
```

Select project you intend to build in Solution's Explorer :

- **Core** : Lib-armadito\libarmadito
- **Module clamav** : modules\clamav_a60
- **Module PDF** : modules\modulePDF
- **Module H1** : modules\moduleH1

Then, launch the build (Run).

Out folder could be one of these :

```
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\Debug
```

or

```
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\Release
```

If build has been successful, you should have these files :

Core :

```
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\conf\armadito.conf
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\glib-2-vs12.dll
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\gmodule-2-vs12.dll
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\gthread-2-vs12.dll
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\libarmadito.dll
```

Module clamav :

```
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\modules\clamav_a60.dll
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\libclamav.dll
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\libeay32.dll
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\ssleay32.dll
```

Module PDF :

```
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\modules\modulePDF.dll
```

Module H1 :

```
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\modules\moduleH1.dll
```

Armadito Windows Driver

Armadito Windows Driver is responsible of on-access protection of Armadito antivirus.

Prerequisites

- Microsoft Visual Studio 2013 (Community edition or more)
- Armadito windows dependencies archive (deps.zip)
- Windows Driver Kit 8.1

To get Windows Driver Kit 8.1 : <<https://www.microsoft.com/en-us/download/details.aspx?id=42273>>

Warning: Windows Driver Kit 8.1 goes **only** with MS Visual Studio 2013. You must get the WDK compatible to your Visual Studio version.

Driver Signing

Add your certificate to local store

- Open the Certificate Manager Tool (certmgr.msc)
- Go to **Certificates - Actual User > Personal > Certificates**
- Right-click on the folder and choose **All tasks > Import**
- Then, follow the assistant to import your certificate.

Sign with your certificate

- Open Armadito-av solution in Visual Studio.
- Right-click on project **ArmaditoGuard** and select **Properties**.
- Go to **Configuration Properties > Driver Signing > General**.
- **Sign mode > Product Sign**.
- **Production Certificate > Select from store** and select your certificate previously added.
- Repeat the previous steps for the project **ArmaditoGuard Package**.

Build

Open the armadito-av VS solution at location :

```
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\Armadito-AV.sln
```


Firstly, select **DriverArmaditoGuard** project in Solution Explorer and build it.

Then, select **Driver\ArmaditoGuard Package** project in Solution Explorer and build it.

Finally, select **Setup\ArmaditoGuard-setup** project in Solution Explorer and build it.

Out

Out folder could be one of these :

```
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\Debug
```

or

```
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\Release
```

If build has been successful, you should have this file :

```
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\driver\armaditoguard.cat  
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\driver\armaditoguard.inf  
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\driver\armaditoguard.sys  
SOMEWHERE\armadito-av\build\windows\VS12\Armadito-AV\out\[build_mode]\driver\ArmaditoGuard-setup.exe
```

Armadito gui

Danger: Graphical user interface is under redesign and new version is not yet ported to Windows. That will be fixed as soon as possible.

Installing prerequisites

The prerequisites are:

- node.js
- git (needed by bower)
- bower

Installing node and bower

First, download node.js from <https://nodejs.org/en/download/>, using either the .msi or the .exe installer, at your choice.

During installation, default configuration choices are OK.

Once node is installed, launch a command line.

Checking installation:

```
$ npm --version  
2.15.1
```

Then install bower using:

```
$ npm install -g bower
```

Install git

To use bower, you must first install git.

git for windows is available here : <https://git-for-windows.github.io/>

Checking installation:

```
$ git version
2.8.1.windows.1
```

Installing modules in source tree

Run bower from **armadito-web-ui** directory to install the needed modules:

```
$ cd SOMEWHERE/armadito-web-ui
$ bower install
$ npm install
```

This should output a lot of messages

Running the interface

First, the Armadito daemon must be launched.

Debugging the interface

Once the interface is launched:

- right-click in the window to display debug menu and select “Inspect” or tap F12
- in the inspector window, select the “console” tab

Build with grunt

Install grunt :

```
npm install -g grunt-cli
```

Run *grunt* for building and *grunt serve* for preview.

You can use “-force” if you want to build with warnings.

Note: This project is generated with [yo angular generator] version 0.15.1.

Installation

Installation on Linux

Installation from sources

When installing Armadito AV from sources (either tarball or git clone), you must first build it. Refer to section **Compilation > Compilation on Linux** of this documentation for detailed instructions.

After proper configuration and build of the different parts (core, modules, gui), each part is simply installed by the following command:

```
$ make install
```

This will install libraries, tools, header files... in the subdirectories of the **PREFIX** directory defined at configure time.

This command must be repeated for each directory (core/, modules*//, gui/).

Care must be take to configure each part with the same **prefix** so that the different components are installed at their respective locations.

Installation from packages

Ubuntu distributions

Packages for Ubuntu distributions are available at:

URL: <<https://launchpad.net/~armadito/+archive/ubuntu/armadito-av>>

```
sudo add-apt-repository ppa:armadito/armadito-av
```

Note that this PPA is experimental and that graphical user interface is not yet packaged in a clean way.

Installation on Windows

Prerequisites

Packages redistribuables Visual C++ for Visual Studio 2013:

- vc_redist_x64.exe

- vcredist_x86.exe

URL : <<https://www.microsoft.com/fr-fr/download/details.aspx?id=40784>>

Installation with MSI installer

1. Download and install prerequisites (see previous section)
2. Download armadito-av msi:
 - Armadito-db-setup-0.10.0.msi
 - Armadito-setup-0.10.0.msi
3. Install armadito-av modules' databases :
Launch the installer Armadito-db-setup-0.10.0.msi
4. Install armadito-av (analysis service + graphical user interface)
Launch the installer Armadito-setup-0.10.0.msi

Installation from Sources

1. Build the armadito-av sources from the Visual Studio solution project.

Follow instructions in **Compilation > Compilation on Windows** section of this documentation.

2. Copy module's databases files in repository:

```
SOMEWHERE\\armadito-av\\build\\windows\\VS\\Armadito-AV\\out\\[build_mode]\\modules\\DB
```

3. Install the driver by executing the following command in a prompt as administrator :

```
ArmaditoGuard-setup --install
```

4. Install the analysis service by executing the following command in a prompt as administrator :

```
ArmaditoSvc --installBoot (service started at system start)
```

or

```
ArmaditoSvc --install (service started on demand)
```

Configuration

Configuration on Linux

Armadito AV's configuration on Linux is stored in two files :

- `/etc/armadito/armadito.conf`
- `/etc/armadito/conf.d/on-access-linux.conf`

Note: If you have compiled from sources, these configuration files are in your PREFIX directory.

Warning: Configuration presented in this document is used for illustration purposes only.

On-demand scan

You are able to configure how on-demand scan works in `/etc/armadito/armadito.conf` :

```
[on-demand]
white-list-dir = "/boot"; "/dev"; "/etc"; "/proc"; "/run"; "/sys"; "/var"
mime-types="*"
modules="clamav"; "moduleH1"
max-size = 10048576
```

- **white-list-dir** : list of directories excluded from on-demand scan.
- **mime-types** : MIME types of files scanned during on-demand scan.
- **modules** : Modules used by on-demand scan.
- **max-size** : Maximum size of scanned files during on-demand scan.

On-access scan

Linux Armadito AV's on-access scan mainly relies on fanotify API. You can find further information on how it works by reading official man page : [fanotify man7](#).

Configuring on-access scan can be done by modifying `/etc/armadito/conf.d/on-access-linux.conf` :

```
[on-access]
enable=1
enable-permission=1
enable-removable-media=1
mount="/home"
directory="/var/tmp"; "/tmp"
white-list-dir = "/bin"; "/boot"; "/dev"; "/etc"; "/lib"; "/lib32"; "/lib64"
mime-types = "application/x-executable"; "application/pdf"; "application/zip"
modules = "clamav"
max-size = 10048576
```

- **enable** : enable (1) or disable (0) on-access scan.
- **enable-permission** : enable (1) or disable (0) permission check.
 - If **enabled**, files detected as malicious will be blocked by Armadito AV.
 - If **disabled**, files detected as malicious will only be notified.
- **enable-removable-media** : enable (1) or disable (0) removable media monitoring.
 - If **enabled**, removable media mount points will be added on the fly to the monitoring list.
- **mount** : list of directories that will be monitored by mount points. I.e. using FAN_MARK_MOUNT.
- **directory** : list of directories that will be monitored by recursively marking all subdirectories.
- **white-list-dir** : list of directories excluded from on-demand scan.
- **mime-types** : MIME types of files scanned during on-demand scan.
- **modules** : Modules used by on-demand scan.
- **max-size** : Maximum size of scanned files during on-demand scan.

Virus Alerts

When a virus is detected by Armadito AV, an alert report is generated and stored in a defined location. This can be configured by modifying `/etc/armadito/armadito.conf` :

```
[alert]
alert-dir = "/var/spool/armadito"
```

- **alert-dir** : directory where scan alerts will be stored.

Quarantine

To isolate infected files, Armadito AV can put detected files in quarantine. `/etc/armadito/armadito.conf` contains configuration about quarantine :

```
[quarantine]
enable = 0
quarantine-dir = "/var/lib/armadito/quarantine"
```

- **enable** : enable (1) or disable (0) quarantine.
- **quarantine-dir** : directory where will be moved files putted in quarantine.

Configuration on Windows

Armadito AV's configuration on Windows can be modified in the following file :

- `<install_dir>\Armadito-av\conf\armadito.conf`

Note: By default, `install_dir` is `C:\Program Files\Teclib`.

On-demand scan

You are able to configure how on-demand scan works in `<install_dir>\Armadito-av\conf\armadito.conf` :

```
[on-demand]
white-list-dir = "C:\\wl-dir1\\"; "C:\\wl-dir2\\"
mime-types="*"
modules="clamav"; "moduleH1"
max-size = 10048576
```

- **white-list-dir** : list of directories excluded from on-demand scan (not yet implemented).
- **mime-types** : MIME types of files scanned during on-demand scan.
- **modules** : Modules used by on-demand scan.
- **max-size** : Maximum size of scanned files during on-demand scan.

On-access scan

You are able to configure how on-access scan works in `<install_dir>\Armadito-av\conf\armadito.conf` :

```
[on-access]
enable = 0
mime-types="*"
modules="clamav"
```

- **enable** : enable (1) or disable (0) on-access scan.
- **mime-types** : MIME types of files scanned during on-access scan.
- **modules** : Modules used by on-access scan.

Virus Alerts

Not yet implemented.

Quarantine

To isolate infected files, Armadito AV can put detected files in quarantine. `<install_dir>\Armadito-av\conf\armadito.conf` contains configuration about quarantine :

```
[quarantine]
enable = 0
quarantine-dir = "quarantine"
```

- **enable** : enable (1) or disable (0) quarantine.
- **quarantine-dir** : sub directory where will be moved files put in quarantine.

Contribute

As Armadito is an open source antivirus project, we greatly invite developers, engineers, partners, customers or anyone eager to boost technologies to collaborate in the software evolution.

By becoming part of a Teclib's community, you share your ideas and tips, exchange your knowledge and expertise with other members in order to fasten the development of Armadito's solution.

You can participate to the project in different ways, such as:

- installing Armadito antivirus
- submitting bug reports
- requesting new features
- developing your own scan module

Licensing

Armadito antivirus can be divided in subprojects, which are the following :

- Armadito core
- Armadito web ui
- Armadito systray ui
- Armadito windows driver
- Armadito module clamav
- Armadito module PDF
- Armadito module H1

Armadito core

- **Description** : Core library in C language.
- **License** : LGPLv3
- **Github** : [armadito-av](#)

Armadito web ui

- **Description** : Graphical User Interface with angularJs.
- **License** : GPLv3
- **Github** : [armadito-web-ui](#)

Armadito systray ui

- **Description** : System tray integration.
- **License** : GPLv3
- **Github** : [armadito-systray-ui](#)

Armadito windows driver

- **Description** : On-access scan Windows implementation in C using Windows Driver Kit.
- **License** : MS-PL
- **Github** : [armadito-windows-driver](#)

Armadito module clamav

- **Description** : Scan module using clamav library.

- **License** : GPLv3
- **Github** : [armadito-mod-clamav](#)

Armadito module PDF

- **Description** : Scan module used especially for PDFs analysis.
- **License** : GPLv3
- **Github** : [armadito-mod-pdf](#)

Armadito module H1

- **Description** : Heuristic scan module focusing on ELF/PE analysis. Based on work by the DAVFI project (<http://www.davfi.fr/>)
- **License** : GPLv3
- **Github** : [armadito-mod-h1](#)

FAQ

Note: Feel free to ask on forum.armadito.org if you have any question.

Screenshots

Some screenshots :

Status board

ARMADITOANTIVIRUS ✔ Your computer is protected Parameters Statistics ?

STATUS

On-access scan :

Databases update : ▲

Last update : ✔ ↻
08-12-2016 8:36

Module	Date	Status
CLAMAV	08-12-2016 8:36	Up-to-date
MODULEH1	01-09-2014 9:30	Up-to-date

SCAN

JOURNAL

Scan

ARMADITOANTIVIRUS ✔ Your computer is protected Parameters Statistics ?

On-demand scan CUSTOM ▼ Start

124 Scanned **110** Malicious **0** Suspects

Scanning file : /home/vhamon/MalwareStor...184AF8B_report.pdf_.pdf

100%

Threats	Files
Pdf.Dropper.Agent-1507060	/home/vham...PDF_931FE142211DB71D4A101D41175C2292.pdf
Pdf.Dropper.Agent-1506680	/home/vham...PDF_683A3CE69FF7D521EF24B37DED206E64.pdf
Pdf.Dropper.Agent-1507058	/home/vham...PDF_851D8956146457569998D9F6E002C127.pdf
Pdf.Exploit.Agent-35955	/home/vham...PDF_25C8BFA4E24CC20CC56ECA4E3EB57673.tar
Pdf.Dropper.Agent-1506743	/home/vham...PDF_954F611F7A6D77050394D3C03FD05F77.pdf
Pdf.Dropper.Agent-1507026	/home/vham...PDF_975D4C98A7FF531C26AB255447127EBB.pdf
Pdf.Dropper.Agent-1506697	/home/vham...PDF_C0522399C51F160B59139838CE96AF47.pdf

STATUS

SCAN

JOURNAL

Journal

8.1. Status board ✔ Your computer is protected Parameters Statistics 29

ARMADITOANTIVIRUS

JOURNAL AI FRTS QUARANTINE

Forum forum.armadito.org

IRC #armadito at irc.freenode.net