Product Guide

# VirusScan® for UNIX

version 4.40.0

**McAfee®**
System Protection

**Industry-leading intrusion prevention solutions**

McAfee®

# Contents

# 1 Introducing VirusScan® for UNIX

VirusScan® for UNIX detects and removes viruses on UNIX-based systems. The scanner runs from a command-line prompt, and provides an alternative to scanners that use a graphical user interface (GUI). Both types of scanner use the same anti-virus software.

The scanner acts as an interface to the powerful anti-virus scanning engine — the engine common to all our anti-virus products.

Although a few years ago, the UNIX operating system was considered a secure environment against potentially harmful software, it is now seeing more occurrences of software specifically written to attack or exploit security holes in UNIX-based systems. Increasingly, UNIX-based systems interact with Windows-based computers, and although viruses written to attack Windows-based systems do not directly attack UNIX systems, the UNIX system can unknowingly harbor these viruses, ready to infect any client that connects to it.

When installed on your UNIX systems, VirusScan® for UNIX becomes an effective solution against viruses, Trojan-horse programs, and other types of potentially harmful software.

The command-line scanner enables you to search for viruses in any directory or file in your computer "on demand" — in other words, at any time. The command-line scanner also features options that can alert you when they detect a virus or take a variety of automatic actions.

When kept up-to-date with the latest virus-definition (DAT) files, the scanner is an important part of your network security. We recommend that you set up an anti-virus security policy for your network, incorporating as many protective measures as possible.

The following topics are included in this section:

- What's new in this release
- Using this guide
- Resources

# What's new in this release

This release of VirusScan® for UNIX includes the following new features or enhancements:

- Improved support for new packers.
- Improved support for ARC compressed files, using a new option, `--showcomp`. See page 25.

# Using this guide

This guide provides information on installing, configuring and using your product. The following topics are included:

- *Introducing VirusScan® for UNIX* on page 4
  An overview of the product, including a description of new or changed features; an overview of this guide; McAfee contact information.
- Detailed instructions for installing the software.
- Descriptions of product features.
- Detailed instructions for configuring and deploying the software.
- Procedures for performing tasks.

## Audience

This information is intended primarily for two audiences:

- Network administrators who are responsible for their company's anti-virus and security program.
- Users who are responsible for updating virus definition (DAT) files on their workstation, or configuring the software's detection options.

## Conventions

This guide uses the following conventions:

| | |
|---|---|
| **Bold Serif** | All words from the user interface, including options, menus, buttons, and dialog box names.<br>Example:<br>Type the **User** name and **Password** of the desired account. |
| `Courier` | The path of a folder or program; a web address (URL); text that represents something the user types exactly (for example, a command at the system prompt).<br>Examples:<br>The default location for the program is:<br>`C:\Program Files\McAfee\EPO\3.5.0`<br>Visit the McAfee web site at:<br>`http://www.mcafee.com`<br>Run this command on the client computer:<br>`C:\SETUP.EXE` |
| *Italic* | For emphasis or when introducing a new term; for names of product documentation and topics (headings) within the material.<br>Example:<br>Refer to the *VirusScan Enterprise Product Guide* for more information. |
| <TERM> | Angle brackets enclose a generic term.<br>Example:<br>In the console tree under **ePolicy Orchestrator**, right-click <SERVER>. |
|  | **Note:** Supplemental information; for example, an alternate method of executing the same command. |
|  | **Tip:** Suggestions for best practices and recommendations from McAfee for threat prevention, performance and efficiency. |
|  | **Caution:** Important advice to protect your computer system, enterprise, software installation, or data. |
|  | **Warning:** Important advice to protect a user from bodily harm when interacting with a hardware product. |
|  | **New:** New or redesigned feature or option of this release of the product. |

# Resources

McAfee® products denote years of experience, and commitment to customer satisfaction. The McAfee PrimeSupport® team of responsive, highly skilled support technicians provides tailored solutions, delivering detailed technical assistance in managing the success of mission critical projects — all with service levels to meet the needs of every customer organization. McAfee Research, a world leader in information systems and security research, continues to spearhead innovation in the development and refinement of all our technologies.

Refer to the following sections for additional resources:

- Getting product information
- Product services
- Contact information

## Getting product information

Unless otherwise noted, the product documentation is provided as Adobe Acrobat .PDF files available on the product CD or from the McAfee download site.

**Product Guide** — This guide. Product introduction and features, detailed instructions for configuring the software, information on deployment, recurring tasks, and operating procedures.

**Help** — Product information in the Help system that is accessed from within the application on its "man" pages.

**Release Notes^** — *ReadMe.* Product information, resolved issues, any known issues, and last-minute additions or changes to the product or its documentation.

**Contacts^** — Contact information for McAfee services and resources: technical support, customer service, Security Headquarters (AVERT Anti-Virus & Vulnerability Emergency Response Team), beta program, and training. This file also includes phone numbers, street addresses, web addresses, and fax numbers for company offices in the United States and around the world.

**License** — The McAfee License Agreement booklet that includes all of the license types you can purchase for your product. The License Agreement sets forth general terms and conditions for the use of the licensed product.

**^** Text files included with the software application and on the product CD.

# Product services

The following services are available to help you get the most from your McAfee products:

- Beta program
- HotFixes and Patches
- Product "end-of-life" support

### Beta program

The McAfee beta program enables you to try our products before full release to the public — you can learn about and test new features for existing products, as well as try out entirely new products. This program can help you test and implement updated and new features earlier, and in a safe environment. You get the chance to suggest new product features, as well as deal directly with McAfee engineering staff.

To find out more, visit:

http://www/mcafeesecurity.com/us/downloads/beta/mcafeebetahome.htm

### HotFixes and Patches

HotFixes and Patches are released with updated files, drivers, executables, etc., between the major releases of a product. To access the latest HotFixes and Patches, visit:

http://www.mcafeesecurity.com/us/downloads/updates/hotfixes.asp

### Product "end-of-life" support

Your anti-virus software must be kept up-to-date to remain effective against viruses and other potentially harmful software. It is important to update the virus definition (DAT) files regularly. To enable the software to counter the continuing threat, we often make architectural changes to the way that the DAT files and virus-scanning engine work together. It is therefore important that you update your engine when a new version is released. An older engine will not catch many of the new emerging threats.

When we release a new engine, we announce the date after which the existing engine will no longer be supported. For information on our product "end-of-life" policy and for a full list of supported engines and products, visit:

http://www.mcafeesecurity.com/us/products/mcafee/end_of_life.htm

# Contact information

## Technical Support

| | |
|---|---|
| Home Page | http://www.mcafeesecurity.com/us/support/technical_support |
| KnowledgeBase Search | https://knowledgemap.nai.com/phpclient/homepage.aspx |
| PrimeSupport Service Portal * | https://mysupport.nai.com |

## McAfee Beta Program

| | |
|---|---|
| | http://www.mcafeesecurity.com/us/downloads/beta/mcafeebetahome.htm |

## Security Headquarters — AVERT: Anti-virus & Vulnerability Emergency Response Team

| | |
|---|---|
| Home Page | http://www.mcafeesecurity.com/us/security/home.asp |
| Virus Information Library | http://vil.nai.com |
| AVERT WebImmune, * Submitting a Sample | https://www.webimmune.net/default.asp |
| AVERT DAT Notification Service | http://vil.mcafeesecurity.com/vil/join-DAT-list.asp |

## Download Site

| | |
|---|---|
| Home Page | http://www.mcafeesecurity.com/us/downloads/ |
| DAT File and Engine Updates | http://www.mcafeesecurity.com/us/downloads/updates/default.asp |
| | ftp://ftp.mcafeesecurity.com/pub/antivirus/datfiles/4.x |
| Product Upgrades * | https://secure.nai.com/us/forms/downloads/upgrades/login.asp |

## Training

| | |
|---|---|
| On-Site Training | http://www.mcafeesecurity.com/us/services/security/home.htm |
| McAfee University | http://www.mcafeesecurity.com/us/services/education/mcafee/university.htm |

## Customer Service

| | |
|---|---|
| E-mail | https://secure.nai.com/us/forms/support/request_form.asp |
| Web | http://www.mcafeesecurity.com/us/index.asp |
| | http://www.mcafeesecurity.com/us/support/default.asp |
| US, Canada, and Latin America toll-free: | **+1-888-VIRUS NO** or **+1-888-847-8766** |
| | Monday – Friday, 8 a.m. – 8 p.m., Central Time |

For additional information on contacting McAfee — including toll-free numbers for other geographic areas — see the Contact file that accompanies this product release.

* Logon credentials required.

# 2 Installing VirusScan® for UNIX

We distribute the VirusScan® for UNIX software in two ways — on a CD, and as an archived file that you can download from our web site or from other electronic services.

After you have downloaded a file or placed your disk in your CD drive, the installation steps are the same for each type of distribution version.

Review the *Installation requirements* on page 11 to verify that the software will run on your system, then follow the installation steps.

## About the distributions

VirusScan® for UNIX software comes in several distribution versions, one for each supported operating system.

- AIX 4.3.x, 5.0L, 5.1L and 5.2L with all recommended patches installed.
- FreeBSD 4.x for Intel (32-bit).
- Hewlett-Packard HP-UX 11.x and HP-UX 11i, with all recommended patches installed.
- Linux for Intel (32-bit) 2.2 and 2.4 production kernels with libc6 (glibc) and the stdc++ library version 2.8, as present in older Linux distributions. The 2.6 kernel is not supported.
- Linux for Intel (32-bit) 2.4 and 2.6 production kernels with libc6 (glibc) and the stdc++ library version 5, as present in newer Linux distributions, such as Red Hat 9 and SuSE 8.2/9.x. The product has been optimized for Pentium 4 but is fully compatible with all Intel Pentium processors.
- Sun Microsystems Solaris for SPARC architecture, versions 7, 8 and 9 with all recommended patches installed.

For current information about the distribution versions, refer to the Release Notes.

If you install VirusScan® for UNIX software from CD, each version is in its own directory. Each distribution has its own installation script.

# Installation requirements

To install and run the software, you need the following:

- The correct version of the UNIX distribution that you require, installed and running correctly on the target computer. See *About the distributions* on page 10 for information.
- 10MB of free hard disk space for a full installation.
- A minimum of 64MB RAM is required, 128MB IS recommended.
- A CD drive, if you are not downloading the software from a web site.

### Other recommendations

- To install the software and perform on-demand scan operations of your file system, we recommend that you have root account permissions.
- To take full advantage of the regular updates to DAT files from our web site, you need an Internet connection, either through your local area network, or via a high-speed modem and an Internet Service Provider.

# Installing the software

This example shows how to install the software on the Solaris distribution. To install other distributions, substitute the correct file name (for example vsun4400.tar.Z) where the example specifies the distribution file.

### To start the installation script:

1 Download the appropriate VirusScan® for UNIX software distribution from our web site or insert the installation CD.

If you are using the installation CD to obtain the software, you can mount the CD on to the file system.

2 Copy the distribution file to a directory on your system.

We recommend that you use a separate (possibly a temporary) directory — not the directory where you intend to install the software.

3 Change the directory to that containing the distribution file. Use `cd`.

4 Type this line at the command prompt to decompress the file:

```
zcat distribution-file | tar -xf -
```

Here, *`distribution-file`* is the file you copied in Step 2.

5 Type this line at the command prompt to execute the installation script:

```
./install-uvscan installation-directory
```

Here, the *`installation-directory`* is the directory where you want to install the software.

If you do not specify an installation directory, the software is installed in `/usr/local/uvscan`.

If the installation directory does not exist, the installation script asks whether you want to create it. If you do not create the installation directory, the installation cannot continue.

6 The installation script asks whether you want to create symbolic links to the executable, the shared library and the man page. Type `Y` to create each link, or `N` to skip the step.

We recommend that you create these links. Otherwise, you will need to set one of the following environment variables to include the installation directory:

**Table 2-1 Environment variables**

| Distribution | Variable |
|---|---|
| IBM AIX | `LIBPATH` |
| FreeBSD | `LD_LIBRARY_PATH` |
| HP-UX | `SHLIB_PATH` |
| Linux | `LD_LIBRARY_PATH` |
| Sun Solaris | `LD_LIBRARY_PATH` |

The program also looks in the `/usr/lib` or `/lib` directory or the current directory for the shared library.

The installation program copies the program files to your hard disk, then scans your home directory.

If the software discovers a virus, see *Handling viruses* on page 19 to learn about the actions you can take.

If the installation fails, see *Troubleshooting during installation* to learn about possible errors and suggested courses of action.

## Troubleshooting during installation

The following table lists the most common error messages returned if the installation fails. The table also suggests a likely reason for the error and recommends any solutions.

**Table 2-2 Error messages**

| Error | Cause or action |
|---|---|
| `Failed to create install_dir` | Verify that you have permission to create the installation directory. |
| `Cannot write to install_dir` | Verify that you have permission to write to the installation directory. |
| `The install_dir exists, but is not a subdirectory` | Choose another installation directory. |
| `<file> is missing` | The file might not exist. |
| `<file> is not correct` | The file did not install correctly. |

# Testing your installation

After it is installed, the program is ready to scan your system for infected files. You can run a test to determine that the program is installed correctly and can scan properly for viruses. The test was developed by EICAR, a coalition of anti-virus vendors headquartered in Europe, as a method for testing any anti-virus software installation.

**To test your installation:**

1 Open a standard text editor, then type the following line:

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

The line must appear as *one line* in the window of your text editor.

2 Save the file with the name EICAR.COM. The file size will be 68 or 70 bytes.

3 Type the following command to scan the EICAR.COM file:

```
uvscan -v eicar.com
```

When the program examines this file, it reports finding the EICAR test file, but you will not be able to clean or repair it.

The EICAR test file *does not contain a virus* — it cannot spread or infect other files, or otherwise harm your system.

4 When you have finished testing your installation, delete the test file to avoid alarming other users.

If the software appears not to be working correctly, check that you have Read permissions on the test file.

## Troubleshooting when scanning

The following table lists the most common error messages returned if the `uvscan` program fails when scanning. The table also suggests a likely reason for the error and recommends possible solutions.

**Table 2-3 Program messages**

| Program message | Remedy |
|---|---|
| `Cannot find shared object` | ■ AIX — Install the correct version of -xlC.rte. The program does not run on versions earlier than 4.0<br>■ HP-UX — Install the aCC run-time patch.<br>■ Linux — Install LIBC6; LIBC5 is not supported. |
| `Unable to find shared library` | Set the appropriate environment variable:<br>■ For AIX, use `LIBPATH`.<br>■ For HP-UX, use `SHLIB_PATH`.<br>■ For Solaris, FreeBSD and Linux, use `LD_LIBRARY_PATH`. |
| `Cannot execute: permission denied` | Check the file permissions. Incorrect file permissions can prevent the program running correctly. All executables (including the shared libraries) must have read and execute permissions (`r_x`), but we recommend `rwxr_xr_x`<br>All DAT files must have read permissions. |
| `Missing or invalid DAT files` | Re-install the DAT files. |
| `The program has been altered; please replace with a good copy` | Re-install from the original media; the program might be infected. |

# Removing the program

A script is installed at the same time as the VirusScan® for UNIX software, which enables you to remove the product quickly and easily.

### To remove the product from your system:

1 Run the script `uninstall-uvscan`, which is in the VirusScan® for UNIX program directory. For example, type the following command at the command prompt:

```
/usr/local/uvscan/uninstall-uvscan
```

2 Delete the script `uninstall-uvscan` from the program directory to remove the program completely from your system.

If you created your own links to the program and a shared library path when you installed the software, you must remove those links yourself.

If you are an administrator, ensure that your users cannot accidentally remove their VirusScan® for UNIX software.

Removing the software leaves your computer unprotected against virus attack. Remove the product only when you are sure that you can upgrade quickly to a new version.

# 3 Using VirusScan® for UNIX

VirusScan® for UNIX provides virus scanning from a command line. This section describes how to use its features and customize the program to meet your needs.

The following features offer optimum protection for your computer and network:

- On-demand scanning options let you start a scan immediately or schedule automatic scans.
- Advanced heuristic analysis detects previously unknown macro viruses and program viruses.
- Updates to virus definition files and upgrades to program components ensure that the program has the most current scanning technology to deal with viruses as they emerge.

Later sections in this guide describe each of these features in detail.

## Running an on-demand scan

You can scan any file or directory on your file system from the command line by adding options to the basic command.

Only the Intel-based FreeBSD and Linux distributions of the VirusScan® for UNIX program can scan for boot-sector viruses.

When executed without options, the program displays a brief summary of its options. When executed with only a directory name specified, the program scans every file in that directory only, and issues a message if any infected files are found. The options fall into the following main groups:

- **Scanning options —** determine how and where the scanner looks for infected files. See page 23.
- **Response options —** determine how the scanner responds to any infected files. See page 26.
- **General options —** determine how the scanner reports its scanning activities. See page 27.

Each group of options appears in its own table with a description of its function. See *Choosing the options* on page 22 for details.

## Command-line conventions

Use the following conventions to add options to the command line:

- Type each option in lower case and separate each with spaces.
- Do not use any option more than once on the command line.
- Follow the syntax correctly. The UNIX operating system is case-sensitive.
- Type single consecutive options as one option. For example, instead of typing this:

  ```
  -c -r --one-file-system
  ```

  you can type this:

  ```
  -cr --one-file-system
  ```

- To start the program, at the command prompt, type:

  ```
  uvscan
  ```

  (This example assumes that the scanner is available in your search path.)

- To have the program examine a specific file or list of files, add the target directories or files to the command line after `uvscan`. You can also create a text file that lists your target files, then add the name of the text file to the command line. See *Configuring scans* on page 17.

  By default, the program examines all files, no matter what their extensions. You can limit your scan by adding only those extensions you want to examine to the command line after the `--extensions` option, or you may exclude certain files from scans with the `--exclude` option. See *Choosing the options* on page 22 for details.

## General hints and tips

The following examples assume that the scanner is available in your search path.

- To display a list of all options, with a short description of their features, type the command:

  ```
  uvscan --help
  ```

- To display a list of all the viruses that the program detects, type the command:

  ```
  uvscan --virus-list
  ```

- To display information about the version of the program, type the command:

  ```
  uvscan --version
  ```

- To scan all subdirectories within a directory with maximum security, type the command:

  ```
  uvscan -r --secure target
  ```

- To ensure maximum protection from virus attack, you must regularly update your DAT files. See *Preventing Infections* on page 31 for details.

# Configuring scans

Instead of running each scan with all its options directly from the command line, you can configure a scan with the options you choose, then save it in a text file as a scan task.

This allows you to run complete scans with ease, and at any time. Your scan task specifies the actions that are performed when a virus is detected.

### To configure a scan:

1 Choose the command options that you want to use.

See *Choosing the options* on page 22 for a description of available options.

2 Type the command options into a text editor just as you might on the command line.

3 Save the text as a file.

4 Type one of these lines at the command prompt:

```
uvscan --load file target

uvscan --config file target
```

Here, *file* is the name of the text file you created, and *target* is the file or directory you want to scan.

If the scanner detects no virus infections, it displays no output.

To learn how to specify the options, see *Command-line conventions* on page 16.

The following examples show how you can configure scans using task files. The examples assume the scanner is available in the search path.

### Example 1

To scan files in the `/usr/docs` directory according to the settings you stored in the task file, `/usr/local/config1`, type the command:

```
uvscan --load /usr/local/config1 /usr/docs
```

The contents of the task file `/usr/local/config1`, are:

```
-m /viruses --ignore-compressed --maxfilesize 4
```

They instruct the scan to move any infected files to `/viruses`, to ignore any compressed files in the target directory, and to examine only files smaller than 4MB.

As an alternative, you can arrange the contents of the task file as separate lines:

```
-m /usr/local/viruses

--ignore-compressed

--maxfilesize 4
```

### Example 2

To scan only files smaller than 4MB and to ignore any compressed files in three separate directories, type the command:

```
uvscan --load /usr/local/config1 --file mylist
```

The contents of the task file `/usr/local/config1`, are:

```
--ignore-compressed

--maxfilesize 4
```

The contents of the other file, `mylist`, are:

```
/usr/local/bin

/temp

/etc
```

## Scheduling scans

You can use the UNIX `cron` scheduler to run automated scans. `Cron` stores the scheduling commands in its `crontab` files. For further information about `cron` and `crontab`, refer to your UNIX documentation or view the Help text, using the commands, `man cron` or `man crontab`.

### Examples

To schedule a scan to run at 18:30 (6:30 p.m.) every weekday, add the following to your `crontab` file:

```
30 18 * * 1-5 /usr/local/bin/uvscan
```

To schedule a scan to run and produce a summary at 11:50 p.m. every Sunday, add the following to your `crontab` file:

```
50 23 * * 0 /usr/local/bin/uvscan --summary
```

To schedule a scan to run on the `work` directory at 10:15 a.m. every Saturday in accordance with options specified in a configuration file `conf1`, add the following to your `crontab` file:

```
15 10 * * 6 /usr/local/bin/uvscan --load conf1 /work
```

To schedule a scan to run at 8:45 a.m. every Monday on the files specified in the file `mylist`, add the following to your `crontab` file:

```
45 8 * * 1 /usr/local/bin/uvscan --f /usr/local/mylist
```

# Handling viruses

If the scanner discovers a virus while scanning, it returns exit code number 13. See *Exit codes* on page 30 for a full description of each code.

To clean infected files or directories, or move them to a quarantine location on your network, you can configure your scanner using one or more response options, which are described in *Response options* on page 26,

The following examples show how you can use these options to respond to a virus attack. The examples assume that the scanner is available in your search path.

### Example 1

To scan and clean all files in the `/usr/docs` directory and all of its subdirectories, type the command:

```
uvscan -cr /usr/docs
```

### Example 2

To scan and clean all files in the `/usr/docs` directory and its subdirectories, but ignore any other file systems that are mounted, type the command:

```
uvscan -cr --one-file-system /usr/docs
```

### Example 3

To scan all files except compressed files in the `/usr/docs` directory and its subdirectories, and to move any infected files to `/viruses`, type the command:

```
uvscan -m /viruses -r --ignore-compressed /usr/docs
```

### Example 4

To scan a file with a name prefixed with "`-`", type the command:

```
uvscan -c -v - -myfile
```

The program scans the named file. It cleans any detected viruses and issues a progress message. This format avoids confusion between the names of the options and the name of the target. Without the "`-`" option, the `uvscan` command appears to have three options and no target:

```
uvscan -c -v -myfile
```

# Using heuristic analysis

An anti-virus scanner uses two techniques to detect viruses: signature matching and heuristic analysis.

A *virus signature* is simply a binary pattern that is found in a virus-infected file. Using information in the DAT files, the scanner searches for those patterns.

However, this approach cannot detect a new virus because its signature is not yet known, therefore the scanner uses another technique — *heuristic analysis*.

Programs, documents or e-mail messages that carry a virus often have distinctive features. They might attempt unprompted modification of files, invoke mail clients, or use other means to replicate themselves. The scanner analyzes the program code to detect these kinds of computer instructions. The scanner also searches for legitimate non-virus-like behavior, such as prompting the user before taking action, and thereby avoids raising false alarms.

In an attempt to avoid being detected, some viruses are encrypted. Each computer instruction is simply a binary number, but the computer does not use all the possible numbers. By searching for unexpected numbers inside a program file, the scanner can detect an encrypted virus. By using these two techniques, the scanner can detect both known viruses and many new viruses and variants. Options that use heuristic analysis include `---analyze`, `--manalyze`, `--panalyze`. See *Scanning options* on page 23.

# Handling an infected file that cannot be cleaned

If the scanner cannot clean an infected file, it renames the file to prevent its use. When a file is renamed, only the file extension (typically three letters) is changed. The following table shows the method of renaming.

**Table 3-1 Renaming infected files**

| Original | Renamed | Description |
|---|---|---|
| `Not v??` | `v??` | File extensions that do not start with *v* are renamed with `v` as the initial letter of the file extension. For example, myfile.doc becomes myfile.voc. |
| `v??` | `vir` | File extensions that start with v are renamed as .vir. For example, myfile.vbs becomes myfile.vir. |
| `vir, v01-v99` | | These files are recognized as already infected, and are not renamed again. |
| <blank> | `vir` | Files with no extensions are given the extension, .vir. |

For example, if an infected file called `bad.com` is found, the scanner attempts to rename the file to bad.vom. However, if a file of that name already exists in the directory, the scanner attempts to rename the file to `bad.vir`, `bad.v01`, `bad.v02`, and so on.

For file extensions with more than three letters, the name is usually not truncated. For example, `notepad.class` becomes `notepad.vlass`. However, an infected file called `water.vapor` becomes `water.vir`.

# Producing reports

The program might take some time to complete a scan, particularly over many directories and files. However, the scanner can keep you informed of its progress, any viruses it finds, and its response to them.

The program displays this information on your screen if you add the `--summary` or `--verbose` option to the command line. To learn more about each option, see *Response options* on page 26.

The `--verbose` option tells you which files the program is examining.

When the scan finishes, the `--summary` option identifies the following:

- How many files were scanned.
- How many files were cleaned.
- How many files were not scanned.
- How many infected files were found.

### Example

In the report information below, both the `--summary` and `--verbose` options were used for scanning files in the `/usr/data` directory.

```
$ uvscan --summary  --verbose /usr/data

Scanning /usr/data/*

Scanning file /usr/data/command.com

Scanning file /usr/data/grep.com

Summary report on /usr/data/*

File(s)

    Total files: ..........       2

    Clean: ................       2

    Not scanned: ..........       0

    Possibly Infected: .....      0
```

# Choosing the options

The following sections describe the options you can use to target your scan:

- *Scanning options*.
- *Response options* on page 26.
- *General options* on page 27.
- *Options in alphabetic order* on page 28.

The descriptions use the following conventions to identify the options or required parameters:

- Short versions of each command option appear after a single dash (-).
- Long versions of each command option, if any, appear after two dashes (--).
- Variables, such as file names or paths, appear in italics within brackets (< >).

To learn how to add these options to the command line, see *Command-line conventions* on page 16.

# Scanning options

Scanning options describe how and where each scan looks for infected files. You can use a combination of these options to customize the scan to suit your needs.

**Table 3-2 Scanning options**

| Option | Description |
|---|---|
| `--afc <size>` | Specify the file cache size.<br>By default, the cache size is 12MB. A larger cache size can improve scanning performance in some cases, unless the computer has low memory. The range of sizes allowed is 8MB to 512MB. Specify the size in megabytes. For example, `--afc 64` specifies 64MB of cache. |
| `--allole` | Check every file for OLE objects. |
| `--analyze`<br>`--analyse` | Use heuristic analysis to find possible new viruses in "clean" files.<br>This step occurs after the program has checked the file for other viruses. See also *Using heuristic analysis* on page 20. |
| `--atime-preserve`<br>`-p`<br>`--plad` | Preserve the last-accessed time and date for files that are scanned.<br>Some backup software archives only changed files, and determines this information from each file's last-accessed date (or 'a-time'). Normally, scanning changes that date. This option will preserve the date, enabling the backup software to work as intended. Sometimes when this option is used, the file date is not preserved; if a file contains a virus, or the scan was started by a user who does not own the file, the file date is changed. |
| `--config <file>` | Run the options specified in *<file>*.<br>You cannot nest configuration files within other configuration files. See also *Configuring scans* on page 17 and *Scheduling scans* on page 18. |
| `-d <directory>`<br>`--dat <directory>`<br>`--data-directory <directory>` | Specify the location of the DAT files — `scan.dat`, `names.dat`, and `clean.dat`.<br>If you do not use this option in the command line, the program looks in the same directory from where it was executed.<br>If it cannot find these data files, the program issues exit code 6. |
| `--exclude <file>` | Exclude the directories or files from the scan as specified in *<file>*. |
| `-e`<br>`--exit-on-error` | Quit and display an error message if an error occurs.<br>The error message indicates the severity of the error. See page 30 for an explanation of exit codes. |
| `--extensions <EXT1[,EXT2,...]>` | Examine files that have the specified extensions.<br>You can specify as many extensions as you want. Separate each with a comma, but without a space. If you choose this option, the program scans only susceptible files, files with execute permissions and those you specify here.<br>To see the list of susceptible files, use the `--extlist` option on page 27. |

**Table 3-2 Scanning options** (Continued)

| Option | Description |
|---|---|
| `--extra <file>` | Specify the full path and file name of any `extra.dat` file.<br>If you do not specify this option in the command line, the program looks in the same directory from where it was executed.<br>If it cannot find this file, the program issues exit code 6. |
| `--fam` | Find all macros, not just macros suspected of being infected.<br>The scanner treats any macro as a possible virus and reports that the file contains macros. However, the macros are not removed. If you suspect that you have an infection in a file, you can remove all macros from the file using the `--fam` and `--cleandocall` or `--dam` options (on page 26) together, although you should only do this with caution. |
| `-f <file>`<br>`--file <file>` | Scan the directories or files as specified in *<file>*. |
| `--floppya`<br>`--floppyb` | Scan the boot sector of the disk in drive A or B.<br>This option is for Intel-based UNIX systems only, namely FreeBSD and Linux. |
| `--ignore-compressed`<br>`--nocomp` | Ignore compressed files.<br>By default, the program scans files saved in these compression formats: ICE, LZEXE, PKLITE, Cryptcom, COM2EXE, Diet, Teledisk, Microsoft Expand and GZIP. This option reduces the scanning time but increases the virus threat because many file types are not scanned.<br>By default, the program scans compressed files. |
| `--ignore-links` | Do not resolve any symbolic links and do not scan the link targets.<br>Normally, the program 'follows' each symbolic link and scans the linked file. |
| `--load <file>` | See `--config` option. |
| `--mailbox` | Scan plain-text mailboxes.<br>These include Eudora, PINE, and Netscape. Most mailboxes will be in MIME format, and therefore the `--mime` option is also required.<br>This option does not clean or rename infected mail items; you must first extract them from the mailbox. |
| `--manalyze`<br>`--manalyse`<br>`--macro-heuristics` | Use heuristic analysis to identify potential macro viruses.<br>This is a subset of `--analyze`. See also *Using heuristic analysis* on page 20. |
| `--maxfilesize <size>` | Examine only those files smaller than the specified size.<br>Specify the file size in megabytes. For example, `maxfilesize 5` means scan only files that are smaller than 5MB. |
| `--mime` | Scan MIME-encoded files.<br>This type of file is not scanned by default. |
| `--noboot` | Do not scan the boot sector. |

**Table 3-2 Scanning options** (Continued)

| Option | Description |
|---|---|
| `--nodecrypt` | Do not decrypt Microsoft Office compound documents that are password-protected.<br>By default, macros inside password-protected compound documents are scanned by employing "password cracking" techniques. If, for reasons of security, you do not require these techniques, use this option. Password cracking does not render the file readable. |
| `--nodoc` | Do not scan Microsoft Office document files.<br>This includes Microsoft Office documents, OLE2, CorelDraw, PowerPoint, WordPerfect, RTF, Visio, Adobe PDF 5, Autodesk Autocad 2000, and Corel PhotoPaint 9 files. |
| `--noexpire` | Do not issue a warning if the DAT files are out of date. |
| `--nojokes` | Do not report any joke programs. |
| `--noscript` | Do not scan files that contain HTML, JavaScript, Visual Basic, or Script Component Type Libraries.<br>This type of file is normally scanned by default. Stand-alone Javascript and Visual Basic Script files will still be scanned. |
| `--one-file-system` | Scan an entire directory tree without scanning mounted file systems, if you use this option in conjunction with the `--sub` option.<br>Normally, the program treats a mount point as a subdirectory and scans that file system. This option prevents the scan from running in subdirectories that are on a different file system to the original directory. |
| `--panalyze`<br>`--panalyse` | Use heuristic analysis to identify potential program viruses.<br>By default, the program scans only for known viruses. The `--panalyze` option is a subset of `--analyze`. See also *Using heuristic analysis* on page 20. |
| `--program` | Scan for potentially harmful applications.<br>Some widely available applications, such as "password crackers" can be used maliciously or can pose a security threat. |
| `-r`<br>`--recursive`<br>`--sub` | Examine any subdirectories in addition to the specified target directory. |
| `--secure` | Examine all files, unzip archive files and use heuristic analysis.<br>This option activates the `--analyze` and `--unzip` options. If the `--selected` and `--extensions` options are in the command line, they are ignored. |
| `--showcomp` | Report any files that are packaged. |
| `-s`<br>`--selected` | Look for viruses in any file that has execute permissions, and in all files that are susceptible to virus infection.<br>By default, all files are scanned. By scanning only files that are susceptible to virus infection, the program can scan a directory faster.<br>To see the list of susceptible files, use the `--extlist` option (page 27). |

**Table 3-2 Scanning options** (Continued)

| Option | Description |
|---|---|
| `--timeout <seconds>` | Set the maximum time to scan any one file. |
| `--unzip` | Scan inside archive files, such as those saved in ZIP, LHA, PKarc, ARJ, TAR, CHM, and RAR formats.<br>If used with `--clean`, this option attempts to clean non-compressed files inside .ZIP files only. No other archive formats can be cleaned.<br>The `--clean` option does not delete or rename infected files within .ZIP files. It does not rename the .ZIP file itself.<br>The program cannot clean infected files found within any other archive format; you must first extract them manually from the archive file in order to clean them. |

## Response options

Response options determine how your scanner responds to a virus infection. You can use a combination of these options to customize the scan. None of the options in Table 3-3 occur automatically. To activate each option, specify it in the command line.

**Table 3-3 Response options**

| Option | Description |
|---|---|
| `-c` | Automatically remove any viruses from infected files. |
| `--clean` | By default, the program states only that infections were found but does not try to clean the infected file. If the program cannot clean the file, it displays a warning message. If you use this option, repeat the scan to ensure that there are no more infections. |
| `--cleandocall` | Delete all macros in a file if an infected macro is found. |
| `--dam` | If you suspect that a file is infected, you can choose to remove all macros from the file to prevent any exposure to a virus. To pre-emptively delete all macros in a file, use this option with `--fam` (on page 24), although you should do this with caution. If you use these two options together, all found macros are deleted, regardless of the presence of an infection. |
| | |
| `--delete` | Automatically delete any infected files that are found. |
| `-m <directory>` | Move any infected files to a quarantine location as specified. |
| `--move <directory>` | When the program moves an infected file, it replicates the full directory path of the infected file inside the quarantine directory so you can determine the original location of the infected file.<br>If you use this option with `--clean`, the program copies the infected files to a quarantine location and tries to clean the original. If the program cannot clean the original, it deletes the file. |
| `--norename` | Do not rename an infected file that cannot be repaired.<br>See *Handling an infected file that cannot be cleaned* on page 20 for information about renaming. |

# General options

General options provide help or give extra information about the scan. You may use a combination of these options to customize the scan. None of the options in Table 3-4 occur automatically. To activate each option, specify it as part of the command line.

**Table 3-4 General options**

| Option | Description |
|---|---|
| `-` | Denote the end of the options and the start of the target to be scanned.<br>This optional feature is particularly useful with file names that are prefixed with "`-`", because it avoids confusion between the options and the target. |
| `--extlist` | Display a list of all file extensions that are susceptible to infection.<br>In other words, those file extensions that are scanned when `--selected` is set. |
| `-h`<br>`--help` | List the most commonly used options, with a short description.<br>For a full description, use `man uvscan`. |
| `--summary` | Produce a summary of the scan.<br>This includes the following:<br>■ How many files were examined.<br>■ How many infected files were found.<br>■ How many viruses were removed from infected files. |
| `-v`<br>`--verbose` | Display a progress summary during the scan.See also *Producing reports* on page 21. |
| `--version` | Display the scanner's version number. |
| | |
| `--virus-list` | Display the name of each virus that the scanner can detect.<br>This option produces a long list, which is best viewed from a text file. To do this, redirect the output to a file for viewing. For full details about each virus, see the Virus Information Library under *Contact information* on page 9. |

# Options in alphabetic order

For convenience, the options are repeated in this section in alphabetic order. For fuller descriptions, see the previous sections.

**Table 3-5 Options in alphabetic order**

| Option | Description | See ... |
|---|---|---|
| - | Denote the end of the options and the start of the target to be scanned. | page 27 |
| `--afc <size>` | Specify the file cache size. | page 23 |
| `--allole` | Check every file for OLE objects. | page 23 |
| `--analyse` | Same as `--analyze`. | page 23 |
| `--analyze` | Use heuristic analysis to find possible new viruses in "clean" files. | page 23 |
| `--atime-preserve` | Preserve the last-accessed time and date for files that are scanned. | page 23 |
| `-c` | Same as `--clean`. | page 26 |
| `--clean` | Automatically remove any viruses from infected files. | page 26 |
| `--cleandocall` | Same as `--dam`. | page 26 |
| `--config <file>` | Run the options specified in <file>. | page 23 |
| `-d <directory>` | Same as `--dat <directory>`. | page 23 |
| `--dam` | Delete all macros in a file if an infected macro is found. | page 26 |
| `--dat <directory>` | Specify the location of the DAT files — scan.dat, names.dat, and clean.dat. | page 23 |
| `--data-directory <directory>` | Same as `--dat <directory>`. | page 23 |
| `--delete` | Automatically delete any infected files that are found. | page 26 |
| `-e` | Same as `--exit-on-error`. | page 23 |
| `--exclude <file>` | Exclude the directories or files from the scan as specified in <file>. | page 23 |
| `--exit-on-error` | Quit and display an error message if an error occurs. | page 23 |
| `--extensions <EXT1[,EXT2,...]>` | Examine files that have the specified extensions. | page 23 |
| `--extlist` | Display a list of all file extensions that are susceptible to infection. | page 27 |
| `--extra <file>` | Specify the full path and file name of any extra.dat file. | page 24 |
| `-f <file>` | Same as `--file <file>`. | page 24 |
| `--fam` | Find all macros, not just macros suspected of being infected. | page 24 |
| `--file <file>` | Scan the directories or files as specified in <file>. | page 24 |
| `--floppya` | Scan the boot sector of the disk in drive A or B. | page 24 |
| `--floppyb` | | page 24 |
| `-h` | Same as `--help`. | page 27 |
| `--help` | List the most commonly used options, with a short description. | page 27 |

**Table 3-5 Options in alphabetic order** (Continued)

| Option | Description | See ... |
|---|---|---|
| `--ignore-compressed` | Ignore compressed files. | page 24 |
| `--ignore-links` | Do not resolve any symbolic links and do not scan the link targets. | page 24 |
| `--load <file>` | Same as `--config <file>`. | page 23 |
| `-m <directory>` | Same as `--move <directory>`. | page 26 |
| `--macro-heuristics` | Same as `--manalyze`. | page 24 |
| `--mailbox` | Scan plain-text mailboxes. | page 24 |
| `--manalyse` | Same as `--manalyze`. | page 24 |
| `--manalyze` | Use heuristic analysis to identify potential macro viruses. | page 24 |
| `--maxfilesize <size>` | Examine only those files smaller than the specified size. | page 24 |
| `--mime` | Scan MIME-encoded files. | page 24 |
| `--move <directory>` | Move any infected files to a quarantine location as specified. | page 26 |
| `--noboot` | Do not scan the boot sector. | page 24 |
| `--nocomp` | Same as `--ignore-compressed`. | page 24 |
| `--nodecrypt` | Do not decrypt Microsoft Office compound documents that are password-protected. | page 25 |
| `--nodoc` | Do not scan Microsoft Office document files. | page 25 |
| `--noexpire` | Do not issue a warning if the DAT files are out of date. | page 25 |
| `--nojokes` | Do not report any joke programs. | page 25 |
| `--norename` | Do not rename an infected file that cannot be repaired. | page 26 |
| `--noscript` | Do not scan files that contain HTML, JavaScript, Visual Basic, or Script Component Type Libraries. | page 25 |
| `--one-file-system` | Scan an entire directory tree without scanning mounted file systems, if you use this option in conjunction with the –sub option. | page 25 |
| `-p` | Same as `--atime-preserve`. | page 23 |
| `--panalyse` | Same as `--panalyze`. | page 25 |
| `--panalyze` | Use heuristic analysis to identify potential program viruses. | page 25 |
| `--plad` | Same as `--atime-preserve`. | page 23 |
| `--program` | Scan for potentially harmful applications. | page 25 |
| `-r` | Same as `--sub`. | page 25 |
| `--recursive` | Same as `--sub`. | page 25 |
| `-s` | Same as`--selected`. | page 25 |
| `--secure` | Examine all files, unzip archive files and use heuristic analysis. | page 25 |
| `--selected` | Look for viruses in any file that has execute permissions, and in all files that are susceptible to virus infection. | page 25 |
| `--showcomp` | Report any files that are packaged. | page 25 |
| `--sub` | Examine any subdirectories in addition to the specified target directory. | page 25 |

**Table 3-5 Options in alphabetic order** (Continued)

| Option | Description | See ... |
|---|---|---|
| `--summary` | Produce a summary of the scan. | page 27 |
| `--timeout <seconds>` | Set the maximum time to scan any one file. | page 26 |
| `--unzip` | Scan inside archive files, such as those saved in ZIP, LHA, PKarc, ARJ, TAR, CHM, and RAR formats. | page 26 |
| `-v` | Same as `--verbose`. | page 27 |
| `--verbose` | Display a progress summary during the scan.See also Producing reports on page 21. | page 27 |
| `--version` | Display the scanner's version number. | page 27 |
| `--virus-list` | Display the name of each virus that the scanner can detect. | page 27 |

# Exit codes

When it exits, VirusScan® for UNIX returns a code to identify any viruses or problems that were found during a scan.

**Table 3-6 Exit codes**

| Code | Description |
|---|---|
| 0 | The scanner found no viruses and returned no errors. |
| 2 | Integrity check on a DAT file failed. |
| 6 | A general problem occurred. |
| 8 | The scanner could not find a DAT file. |
| 12 | The scanner tried to clean a file, and that attempt failed for some reason, and the file is still infected. |
| 13 | The scanner found one or more viruses or hostile objects — such as a Trojan-horse program, joke program, or test file. |
| 15 | The scanner's self-check failed; it may be infected or damaged. |
| 19 | The scanner succeeded in cleaning all infected files. |
| 102 | The scanner quit because the `--exit-on-error` option. was included.<br>This code appears when the scan encounters an unexpected condition; for example, if it cannot open a file or runs out of available memory. The program exits immediately and does not finish the scan. This code occurs only if you specified the `--exit-on-error` option when you started the program. If you did not specify the `--exit-on-error` option, the scanner returns exit code 6. |

# 4 Preventing Infections

VirusScan® for UNIX is an effective tool for preventing infections, and it is most effective when combined with regular backups, meaningful password protection, user training, and awareness of virus threats.

To create a secure system environment and minimize the chance of infection, we recommend that you do the following:

- Install VirusScan® for UNIX software, and other McAfee anti-virus software where applicable.
- Include a `uvscan` command in a `crontab` file.
- Make frequent backups of important files. Even if you have VirusScan® for UNIX software to prevent infections, damage from fire, theft, or vandalism can render your data unrecoverable without a recent backup.

## Detecting new and unidentified viruses

To offer the best virus protection possible, we continually update the virus definition (DAT) files that the VirusScan® for UNIX software uses to detect viruses. For maximum protection, you should regularly retrieve these files.

We offer free online DAT file updates for the life of your product, but cannot guarantee they will be compatible with previous versions. By updating your software to the latest version of the product and updating regularly to the latest DAT files, you ensure complete virus protection for the term of your software subscription or maintenance plan.

## Why do I need new DAT files?

Hundreds of new viruses appear each month. Often, older DAT files cannot detect these new variations. For example, the DAT files with your original copy of VirusScan® for UNIX might not detect a virus that was discovered after you bought the product.

If you suspect you have found a new virus, use WebImmune. See *Contact information* on page 9 for the address.

## Updating your DAT files

The DAT files are contained in a single compressed file. Download the new file from either of the following sources:

- **FTP server.** Open a connection to the ftp site. See *Download Site* under *Contact information* on page 9 for the address.

  Use `anonymous` as your user name and your e-mail address as your password to gain access. Look for a file with the format *dat-nnnn.zip*, where *nnnn* is the DAT version number. For example: dat-4432.zip.

- **Web Site.** Start your browser, then go to the **Downloads** area for the latest file.

The number given to the file changes on a regular basis. A higher number indicates a later version of the DAT files. When you are selecting the latest version of DAT file, ignore any reference to SuperDAT (a self-installing DAT file).You cannot use this type of file with the command-line scanner.

### To use the new DAT files:

1. Create a download directory.
2. Change to the download directory, and download the new compressed file from the source you have chosen.
3. To unpack the DAT files, type the command:

   ```
   tar -xf file
   ```

   Here, *`file`* is the name of the file you downloaded.

4. Type this command to move the DAT files to the directory where your software is installed. Name the file using lower case.

   ```
   mv *.dat installation-directory
   ```

   Here, *`installation-directory`* is the directory where you installed the software. (See *Installing the software* on page 11.)

   Your computer overwrites the old DAT files with the new files. Your anti-virus software will now use the new DAT files to scan for viruses.

## Sample update script for UNIX

The following script is provided only as a suggestion, for you to use and modify to suit your own purposes. It has not been thoroughly tested. Further error checking and password authentication might be required.

The following example shows an update script that gets new DAT files from the FTP site.

This entry must appear in the .netrc file for this script to work:

```
machine ftp.nai.com
login anonymous
password e-mail address
macdef init
cd pub/antivirus/datfiles/4.x
bin
prompt
mget dat-*.tar
close
bye
```

where *e-mail address* is the address of the user who is logging in to the FTP server.

```
#!/bin/sh

# Assume uvscan is installed in the same directory
# as this script.
install_directory=`dirname $0`

# Create a download directory
mkdir /tmp/dat-updates
cd /tmp/dat-updates

# Get the version of the currently installed DAT files
# from the info given by the --version option

current_version=`

$install_directory/uvscan --version |
grep "Virus data file" |
awk '{ print substr($4,2,4) }'`

# Get the new DATs.
# The entry in your .netrc file should take care
# of the downloading.
ftp ftp.nai.com



# Get the version of the new DATs from the file name.
new_version=`echo dat-*.tar | awk '{ print substr($1,5,4) }'`
```

```
# If they are the same age or older
# than the current ones,do not install them.
if [ "$current_version" -ge "$new_version" ]
then
 echo "No new DATs available at this time"
 echo "Currently installed version:  $current_version"
 echo "Version on FTP site:     $new_version"
else
 tar -xf dat-*.tar

# Move them to the install directory, making sure
# that the file name is lower case.

for file in `tar -tf dat-*.tar`
    do
       newfile=`echo $file | tr [A-Z] [a-z]`
       mv ./$file "$install_directory/$newfile"
    done

# Get the current version again and make sure
# that the new DATs installed correctly.
current_version=`

$install_directory/uvscan --version |
grep "Virus data file" |
awk '{ print substr($4,2,4) }'`

if [ ! "$current_version" -eq "$new_version" ]
then
   echo "DAT file updates did not work correctly."
   echo "Please try manually."
   fi

fi

# Delete the directory that you created.
cd /
rm -fr /tmp/dat-updates
```

## Sample update script for Perl

This script is provided only as a suggestion for you to use and modify to suit your own purposes. It has not been thoroughly tested. Further error checking and password authentication might be required.

```
#!/usr/bin/perl -w

# uvscan virus DAT file updater written by
# Michael Matsumura (michael+uvscan@limit.org)
# Version 1.0
#
# Net::FTP is required for operation
# and 'tar' should be in the PATH

use strict;

# Set to the directory uvscan is located/installed in.
my $uvscan_directory = "/usr/local/uvscan";



# Set to the temporary directory to download
# the DAT archive.
my $tempdir = "/tmp/dat-updates";

# Set to email address for anonymous FTP login
my $emailaddress = "root@";

use Net::FTP;

# Define global variables
my ($ftp, @dirlist, $arraywalk, $localver, $serverver, $localfile,
@files, $file);

# Get the local uvscan datfile version
$localver = &checkuvscanver;
print "Currently installed version: ".$localver."\n";

# Create FTP connection
$ftp = Net::FTP->new("ftp.nai.com", Debug => 0);

# Login
$ftp->login("anonymous",$emailaddress);
$ftp->cwd("/pub/antivirus/datfiles/4.x");
$ftp->binary();

@dirlist = $ftp->ls();

foreach $arraywalk (@dirlist) {

  if ($arraywalk =~ /dat-([0-9]+)\.tar/i) {

    $serverver = $1;

    print "Version on ftp.nai.com: ".$serverver."\n";

    if ($serverver > $localver) {

      print "Updating virus data files...\n";
```

```
# Create and then change the working directory to $tempdir
if (!(-d $tempdir)) {
  mkdir($tempdir, 700) or die("ERROR: Couldn't make   temporary
directory: $tempdir");
  }

chdir $tempdir or die("ERROR: Couldn't change directory to tempdir:
   $tempdir");

# Download the DAT file!
$localfile = $ftp->get($arraywalk);
print "Download complete...updating now\n";

# Untar the files, store the names of them into an array

      @files = `tar -xvf $arraywalk`;

      foreach $file (@files) {

# A line break is at the end of each $file...
# chomp that off

        chomp($file);

# Move each file to the uvscan_directory;
# and make sure they are lowercase.

      my $movestring = "mv $file ".$uvscan_directory."/".lc($file);

        print "   ".$movestring."\n";

        system($movestring);

      }

# Make sure that the installation worked,
# by checking if the virus scanner reports
# the same data file version as the one we downloaded.

      if (&checkuvscanver eq $serverver) {

        print "Installation successful\n";

      } else {

        print "Error in installation, please install manually\n";

      }


# Cleanup...
   print "Cleaning up\n";

# Remove downloaded DAT archive
unlink($arraywalk) or die("ERROR: Couldn't delete DAT file:
  $arraywalk");
```

```
# Change to filesys root
# and remove temporary directory
chdir("/");
rmdir($tempdir) or die("ERROR: Couldn't remove tempdir: $tempdir");



    } else {

#

if ($serverver > $localver) {
     print "DAT files are the same..no need to update\n";

   }

# Don't want to continue if there is more than
# one 'dat-[0-9]+.tar' files

   last;

  }
}

$ftp->quit;



# uvscan --version reports...
# "Virus data file 4229 created Oct 16 2002"
# &checkuvscanver returns the version
# of the data files.

sub checkuvscanver {

  if (`$uvscan_directory/uvscan --version` =~ /Virus data file

v([0-9]+) created/) {

    return $1;

  }
}
```

# Index