Automatic updates for Websense data endpoints

Topic 41102 / Updated: 25-Feb-2014

Applies To:	Websense Data Security v7.6, v7.7.x, and v7.8

Endpoint auto-update is a feature that lets a network server push an endpoint installation package to client machines and silently install the package in the background. By doing so, the network server controls the version of the endpoint running on client machines.

Note that the endpoint auto-update feature does not support the initial deployment of the agent — it only supports existing agents. In addition, it does not apply to Linux or Mac endpoints. It works only with Windows endpoint clients.

This document is divided into two sections:

- *Configuring the auto-update server*
- *Auto-update workflow for advanced configuration*

The first section describes how to set up a server to work with the auto-update feature. The second section helps you understand the work flow of the endpoint auto-update process, how the endpoint and update server communicate with each other, and how you can add flexibility to the endpoint auto-update feature with different parameters.

Configuring the auto-update server

The steps provided in this section need to be performed only once. Once this setup is completed, you do not need to repeat this process.

To set up the auto-update system, follow these steps:

1. Install a Web server.
2. Copy and unzip endpoint server-side files to the Web server.
3. Configure the Web server.
4. Deploy an initial endpoint package on your endpoint clients.
5. Deploy an endpoint package on the auto-update server.

Installing Web server

The endpoints that perform automatic updates regularly check with a Web server to determine if they are at the most current version. If the endpoints are not up to date, they try to download a new package from the Web server and install it.

Your Web server can be any server in your network. For best practice, it should be on a different machine than your Websense servers— such as the TRITON management server and secondary Data Security servers. This optimizes performance of the Websense servers and preserves them for future upgrades. It also gives you the flexibility to choose the port numbers, the hardware and operating system, as well as the security hardening mechanisms to be used, without the risk of collision with Websense components.

You can choose any Web server software that meets your needs and configure it on your machine and network, as long as it meets the following requirements:

- It must support file hosting.
- It must support CGI or other server-side scripting language.
- It must have enough hardware resources to handle I/O from all endpoints. Generally, when endpoints are up-to-date, they query the server every 120 minutes, with each query and response being approximately 1 KB. But when endpoints are out-of-date, they try to download the update package, which is typically 100 MB.

  Therefore, a server that supports 1,200 endpoints should expect 10 requests per minute (1200 per 120 minutes). When a new package is available, each request can result in a 100 MB file transfer.

  Note that endpoints retry their communication attempts if the server can't handle the load.
- It must be accessible by the network where the endpoints are installed.
- Its URL must begin with HTTP:// and not HTTPS://. Secure HTTP is not supported.

Later, this document provides instructions on how to use 3 different types of Web servers and provides sample installation instructions for each. See *Configuring your Web server* below for details.

Copying server-side foundation files

When your server is ready, you need to copy the Endpoint Update Server Kit to your Web server machine and unzip the files. To do so:

1. Log on to a machine where Websense Data Security is installed.
2. Locate the zip file **Endpoint Update Server Kit.zip** under the installation folder (%DSS_HOME%).
3. Copy the file to your Web server machine in a location that is accessible by the Web server process — for example, EP_UPDATE_ROOT.
4. Unzip the file.

EP_UPDATE_ROOT should now contain the following subfolders:

- conf
- scripts_windows
- scripts _linux
- data

Configuring your Web server

To configure your Web server, follow these basic steps:

1. Choose a scripts folder to use from **EP_UPDATE_ROOT** (either **scripts_windows** or **scripts_linux**).
2. Create a virtual directory called **/EPUpdate** that is CGI-enabled, and is linked to **EP_UPDATE_ROOT/scripts**.
3. Create another virtual directory called **/EPPackages** that links to **EP_UPDATE_ROOT/data**.

Note that each Web server installation has different configuration steps. Listed below are steps for the 3 most common Web servers:

- *Apache HTTPD on Windows*, page 3
- *Apache HTTPD on Linux*, page 3
- *Microsoft IIS 7.x on Windows 2008 or Server 2008 R2*, page 4

Apache HTTPD on Windows

a. Rename the **EP_UPDATE_ROOT/scripts_windows** folder to **scripts** (EP_UPDATE_ROOT/scripts).

b. Edit the configuration file **EP_UPDATE_ROOT/conf/httpd.conf** with a text editor and replace the string **${EP_UPDATE_ROOT}** with the actual value of EP_UPDATE_ROOT.
Important: Use forward slash (/) characters to separate folders. Do not use back slash characters (\).

c. Locate the text file, **httpd.conf**, in the Apache-HTTPD installation folder. Edit the file and append a single line at its end:
include EP_UPDATE_ROOT/conf/httpd.conf

d. Restart the Apache HTTPD service. Make sure that the service starts up.

For additional information, refer to the installation instructions provided on the Apache Web site for compiling and installation on Windows.

Apache HTTPD on Linux

a. Rename the **EP_UPDATE_ROOT/scripts_linux** folder to **scripts** (EP_UPDATE_ROOT/scripts).

b. Run the following command to make sure **EP_UPDATE_ROOT/scripts/update** has execute permissions:

chmod +x EP_UPDATE_ROOT/scripts/update

c. If your Linux server is running SELinux (Security Enhanced Linux), use the semanage or the chcon command to label the file-type **EP_UPDATE_ROOT/scripts/update** as **httpd_sys_content_t**. To do this, run the following commands as a Linux root user:

- /usr/sbin/semanage fcontext -a -t httpd_sys_content_t EP_UPDATE_ROOT/scripts/update
- /sbin/restorecon EP_UPDATE_ROOT/scripts/update

d. Edit the configuration file **EP_UPDATE_ROOT/conf/httpd.conf** with a text editor, and replace the string **${EP_UPDATE_ROOT}** with the actual value of EP_UPDATE_ROOT.

e. Edit the file **/etc/httpd/conf/httpd.conf**, and append a single line at its end:

include EP_UPDATE_ROOT/conf/httpd.conf

f. Restart the Apache HTTPD service. Make sure that the service starts up.

For additional information, see the Installation instructions provided on the Apache Web site.

Microsoft IIS 7.x on Windows 2008 or Server 2008 R2

1. Click **Start**, point to **Administrative Tools**, and then click **Server Manager** to open the IIS Manager.
2. In the left pane, click on the machine name, then double-click on the option **ISAPI and CGI Restrictions** in the right pane.
   a. Right-click an empty area in the right pane, select **Add**, and fill in the following values:
      - ISAPI or CGI path: **EP_UPDATE_ROOT\scripts_windows\update.bat**
      - Description: **Websense Data Endpoint Auto-Update**
   b. Check the option, **Allow Extention path to execute**.
   c. Click **OK**.
3. On the tree in the left pane, locate the site where you want to host your auto-update server (the default is **Default Web Site**).
4. Right-click the site, choose **Add Virtual Directory**, and enter the following details:
   - Alias: **EPUpdate**
   - Physical path: **EP_UPDATE_ROOT\scripts_windows**
5. Click on the newly created **EPUpdate** virtual folder in the left pane, and double-click on **Handler Mappings** in the right pane.
6. Right-click an empty area in the right pane, choose **Add Module Mapping**, and enter the following values:
   - Request Path: **update.bat**
   - Module: **CgiModule**
   - Executable: (leave empty)

- Name: **Websense Data Endpoint Auto-Update**

7. Right-click anywhere on the site, select the option **Add Virtual Directory**, and enter the following details:
   - Alias: **EPPackages**
   - Physical path: **EP_UPDATE_ROOT\data**

For additional information, refer to the installation instructions provided for Windows Server 2008 or Windows Server 2008 R2 on the IIS Web site.

Deploying the initial endpoint package on your endpoint clients

Use the Websense Endpoint package builder to create an initial installation package for your endpoints. You must deploy this installation package to your endpoint clients yourself, based on your preference, such as Active Directory GPO or Microsoft SMS.

- When creating the package, set up a URL for automatic updates:
  - If you have installed an Apache HTTPD server (Windows or Linux), the URL should be: http://server:port/EPUpdate/update
  - If you have installed an IIS server, the URL should be http://server:port/EPUpdate/update.bat
- Also, set up a schedule for how often the endpoint clients should check for updates.

  See the Deployment and Installation Center for instructions on using the endpoint package builder.

Important

At the completion of any update, you must restart the endpoint to ensure the update takes effect.

Deploying an endpoint package on the auto-update server

Follow these steps to deploy a new package using the auto-update mechanism.

a. Create the package.

   Use the endpoint package builder to create a new package, exactly like you created the initial one. The package builder generates a folder with several installation packages, one per each version of the operating system.

   Note: If you plan to use auto-update frequently, make sure that new packages point to an auto-update server.

b. Add package metadata.

   On the Data Security Management Server, open a command prompt and change to the %DSS_HOME% directory:

   ```
   cd %DSS_HOME%
   ```

   Run the following command (in a single line):

"%DSS_HOME%\python" "%DSS_HOME%\EP_Prepare_Package4Update.py" "Path-to-folder-with-packages"

This command creates a new subfolder called **.private** inside the folder with the generated package. This subfolder contains metadata about the package.

c. Copy the package to the Web server machine.

Copy the entire contents of the generated package folder (along with the metadata) to the Web server machine (into EP_UPDATE_ROOT/data). For example, the win32 installation will be located in EP_UPDATE_ROOT/data/WebsenseEndpoint_32bit.exe.

Be aware that if you copy an older endpoint package to the Web server (inadvertantly or otherwise), the endpoints will download and install the older version.

Now your server is ready. Whenever there is a new Data Endpoint release, copy the updated release binaries to your auto-update server, and the endpoints will update at the next scheduled time.

Important

At the completion of any endpoint update, you must restart the endpoint to ensure the update takes effect.

Auto-update workflow for advanced configuration

Read this section to understand the work flow of the endpoint auto-update process, how the endpoint and update server communicate with each other, and how you can add flexibility to the endpoint auto-update feature.

The endpoint client machines check the management server for configuration updates to the data endpoint profile.

This is how the endpoint auto-update process works:

1. The endpoint sends a GET request URL to the auto-update server with local information that the server needs to identify the client machine.
2. The auto-update server responds (in XML-like format) with information about the version of the endpoint installation package, as well as the desired version of the client endpoint.
3. If the desired version is different from the current client version, the endpoint downloads the installation package from the download URL, that was sent as part of the server's response.
4. After the download is complete, the endpoint checks if it is ready to install the software or not.
   a. If the endpoint is ready, the installation package runs silently in the background.

b. If the endpoint is not ready, it postpones the installation until the client machine is ready.

How the endpoint and update server communicate

When the endpoint sends a GET request URL to the auto-update server, the URL contains many pre-defined parameters.

For example:

<UpdateServer URL=http://download.websense.com/update" />

GET /update?Bits=64bit&Platform=Windows&Domain=websense.com&User=EPUser&SID=xxxxxx&LocalVersion=7.6.1218&LocalDSSVersion=7.6.3.16&ProtocalVersion=1.2&WSCookie=4f3h&DLP=Yes&WEB=Yes&RF=No&CI=No HTTP/1.1

Host: download.websense.com

These parameters provide local information about the client machine. Given below is a list of parameters in the GET request URL sent by the endpoint and their description:

Name	Type	Description
Bits	String	Type of OS: 32- or 64-bit
Platform	String	Windows
User	String	First log on user name
Domain	String	Domain name
SID	String	Session ID of the first log on user
LocalVersion	String	Version number of the local endpoint client
LocalDSSVersion	String	Version number of DSS
ProtocolVersion	String	Protocol version of the proxy server
WSCookie	String	Data received fromthe server
DLP	String	Whether the local machine has Data Endpoint installed or not: **Yes** or **No**
WEB	String	Whether the local machine has Web Endpoint installed or not: **Yes** or **No**
RF	String	Whether the local machine has Remote Filtering Client installed or not: **Yes** or **No**
CI	String	Whether the local machine has the Citrix Integration service installed or not: **Yes** or **No**

Similarly, when the auto-update server returns a string in XML-like format, it also includes many pre-defined parameters. For example,

<?xml version="1.0" encoding="utf-8" ?>

<UpdateServer>

<CurrentVersion=”7.6.1219”>

<CurrentDSSVersion=”7.6.3.17”>

<CheckSum=”A15BCDE9393288EFACDB3493827ABEFD”>

<URL="http://download.websense.com/upgrade/installpackage_1219.exe">

<IncludeEP=”Yes”>

<IncludeDSS=”Yes”>

</UpdateServer>

Given below is a description of the elements in the XML-like file returned by the auto-update server:

Name	Type	Description
CurrentVersion	String	Version number of the designated installation package on server.
CurrentDSSVersion	String	Version number of the designated installation package on the server.
CheckSum	String	MD5 checksumof the designated installation package on the server.
URL	String	The URL of the installation package on server. Maximumsize is 2K.
IncludeEP	String	Whether the installation package should include endpoint software or not: **Yes** or **No**.
IncludeDSS	String	Whether the installation package should include DSS software or not: **Yes** or **No**.

Depending on the response of the update server, endpoints can retrieve the install package and install it silently.