Hvr

Contents

Name

hvr - HVR runtime engine.

Synopsis

hvr [**-E***n*=*v*]... [**-t***x*] [*script* [**-***scropts*] [*scrargs*]]

hvr -r [**-A**] [**-E***n*=*v*]... [**-K***pair*] [**-N**] [**-p***pamsrv*] [**-U***user*]... [**-a***accessxml*]

hvr -s*lb*/ [**-E***n*=*v*]...

hvr -x -a*accessxml*/ [**-E***n*=*v*]... [**-K***pair*]

Description

Command **hvr** is an interpreter for HVR's internal script language. These scripts are generated by HVR itself. Inspection of these scripts can improve transparency and assist debugging, but it is unnecessary and unwise to use the internal script language directly because the syntax is liable to change without prior notice between HVR versions.

If no arguments are supplied or the first argument is '**-**' then input is read from **stdin**. Otherwise *script* is taken as input. If *script* begins with '**.**' or '**/**' it is opened as an absolute pathname, otherwise a search for the hvr script is done in the current directory '**.**' and then in **$HVR_HOME/script**.

Command **hvr** with option **-r** is used to provide an HVR child process on a remote machine. Its validation of passwords at connection time is controlled by options **-A**, **-p**, **-N** and **-U**.

Command **hvr** with option **-x** is used to provide an HVR proxy. For more information, see **Hvrproxy**.

Options

This section describes the options available for command **hvr**.

Parameter	Description

-a*accessxml* Unix & Linux	Access control file. This is an XML file for remote connections (option **-r**) and proxy mode (option **-x**) which controls from which nodes connections will be accepted, and also the encryption for those connections. To enable 2-way SSL authentication the public certificate of the hub should be given with XML **<ssl remote_cert="***mycloud***"/>** inside the **<from/>** element of this access control file. Also the public certificate private key pair should be defined on the hub with **LocationProperties /SslLocalCertificateKeyPair**. In proxy mode (option **-x**) this option is mandatory and is also used to control to which nodes connections can be made using XML **<to/>** tags. If *accessxml* is a relative pathname, then the file should be in **$HVR_HOME/lib** and if a SSL certificate is a relative pathname then the file should be in **$HVR_HOME/lib/cert**.
-A Unix & Linux	Remote HVR connections should only authenticate login/password supplied from hub, but should not change from the current operating system username to that login. This option can be combined with the **-p** option (PAM) if the PAM service recognizes login names which are not known to the operating system. In that case the daemon service should be configured to start the HVR child process as the correct operating system user (instead of **root**).
-E*n=v*	Set environment variable *n* to value *v* for this process and its children.
-K*pair* Unix & Linux	SSL public certificate and private key of local machine. This should match the hub's certificate supplied by **/SslRemoteCertificate**. If *pair* is relative, then it is found in directory **$HVR_HOME/lib/cert**. Value *pair* specifies two files; the names of these files are calculated by removing any extension from *pair* and then adding extensions **.pub_cert** and **.priv_key**. For example, option **-Khvr** refers to files **$HVR_HOME/lib/cert/hvr.pub_cert** and **$HVR_HOME/lib/cert/hvr.priv_key.**
-N Unix & Linux	Do not authenticate passwords or change the current user name. Disabling password authentication is a security hole, but may be useful as a temporary measure. For example, if a configuration problem is causing an 'incorrect password' error, then this option will bypass that check.
-p*pamsrv* UNIX & Linux	Use Pluggable Authentication Module *pamsrv* for login password authentication of remote HVR connections. PAM is a service provided by several operation systems as an alternative to regular login/password authentication, e.g. checking the **/etc/passwd** file. Often **-plogin** will configure HVR child process to check passwords in the same way as the operating system. Available PAM services can be found in file **/etc/pam.conf** or directory **/etc/pam.d**.
-r	HVR child process to service remote HVR connections. On Unix/Linux, the **hvr** executable is invoked with this option by the configured daemon. On Windows, **hvr.exe** is invoked with this option by the HVR Remote Listener Service. Remote HVR connections are authenticated using the login/password supplied for the **connect to HVR on a remote machine** information in the location dialog window.
-s*lbl*	Add label */lbl/* to HVR's internal child co-processes. HVR sometimes uses child co-processes internally to connect to database locations. Value */lbl/* has no effect other than to appear next to the process id in the process table (e.g. from **ps -ef**) so that users can distinguish between child co-processes.
-t*x*	Timestamp prefix for each line. Value *x* can be either **s** (which means timestamps in seconds) or **n** (no timestamp). The default is to only prefix a timestamp before each output line if **stderr** directs to a TTY (interactive terminal).

-U*user*	Limits the HVR child process to only accept connections which are able to supply operating system password for account *user*. This reduces the number of passwords that must be kept secret. Multiple **-U** options can be supplied.
-x	HVR proxy mode. In this mode the HVR process will accept incoming connections a reconnect through to other nodes. This requires option **-a**. For more information, see section **Hvrproxy**.

Example

To run hvr script **foo** with arguments **-x** and **bar** and to redirect **stdout** and **stderr** to file log:

```
$ hvr foo -x bar >log 2>&1
```

Custom HVR Password Validation

When **hvr** is used for remote connections (option **-r**) it must validate passwords. This can be customized if an executable file is provided at **$HVR_HOME/lib/hvrvalidpw**. HVR will then invoke this command without arguments and will supply the login and password as **stdin**, separated by spaces. If **hvrvalidpw** returns with exit code **0**, then the password is accepted.

A password validation script is provided in **$HVR_HOME/lib/hvrvalidpw_example**. This script also has options to manage its password file **$HVR_HOME/lib/hvrpasswd**. To install custom HVR password validation,

1. Enable custom password validation.

   ```
   $ cp $HVR_HOME/lib/hvrvalidpw_example $HVR_HOME/lib/hvrvalidpw
   ```

2. Add option **-A** to **Hvrremotelistener** or to the **hvr -r** command line. This prevents an attempt to change the user. Also change **Hvrremotelistener** or the daemon configuration so that this service runs as a non-root user.
3. Add users to the password file **hvrpasswd**.

   ```
   $ $HVR_HOME/lib/hvrvalidpw newuser # User will be prompted for password
   $ $HVR_HOME/lib/hvrvalidpw -b mypwd newuser # Password supplied on
   command line
   ```

Files

HVR_HOME	
bin	
hvr	HVR executable (Unix and Linux).
hvr.exe	HVR executable (Windows).
hvr_ii*N*.dll	Ingres version *N* shared library (Windows).
hvr_or*N*.dll	Oracle version *N* shared library (Windows).
hvr_ms*N*.dll	SQL Server version *N* shared library (Windows).
lib	
cert	
hvr.priv_key	Default SSL encryption private key, used if **hvr**is supplied with option **-Chvr** or **-Khvr** (instead of absolute path). Must be created with command **hvrsslgen**.

hvr.pub_cert	Default SSL encryption public certificate, used if **hvr**is supplied with option **-Chvr** or **-Khvr**or **/SslRemoteCertficate=hvr** (instead of absolute path). Must be created with command **hvrsslgen**.
ca-bundle.crt	Used by HVR to authenticate SSL servers (FTPS, secure WebDAV, etc). Can be overridden by creating new file *host***.pub_cert** in this same certificate directory. No authentication done if neither file is found. So delete or move both files to disable FTPS authentication. This file can be copied from e.g. **/usr/share/ssl/certs/ca-bundle.crt** on Unix/Linux.
*host***.pub_cert**	Used to override **ca-bundle.crt** for server verification for *host*.
hvr_ii*N***.sl** *or***.so**	Ingres shared library (Unix and Linux).
hvr_or*N***.sl** *or* **.so**	Oracle shared library (Unix and Linux).
hvrpasswd	Password file employed by **hvrvalidpwfile**.
hvrvalidpw	Used by HVR for user authentication.
hvrvalidpwfile	The plugin file for private password file authentication.
hvrvalidpwldap	The plugin file for LDAP authentication.
hvrvalidpwldap.conf	Configuration for LDAP authentication plugin.
hvrvalidpwldap.conf_example	Example configuration file for LDAP authentication plugin.
hvrscripthelp.html	Description of HVR's internal script syntax and procedures.

HVR_CONFIG	
job	HVR scripts generated by **hvrinit**.
files	
[*hubnode*]-*hub-chn-loc*.**logrelease**	Status of HVR log-based capture jobs, for command **hvrlogrelease**.
tmp	Temporary files if **$HVR_TMP** is not defined.
HVR_TEMP	Temporary files for sorting and large objects.