NAME

vlmcsd − a fully Microsoft compatible KMS server

SYNOPSIS

vlmcsd [*options*]

DESCRIPTION

vlmcsd is a fully Microsoft compatible KMS server that provides product activation services to clients. It is meant as a drop-in replacement for a Microsoft KMS server (Windows computer with KMS key entered). It currently supports KMS protocol versions 4, 5 and 6.

vlmcsd is designed to run on POSIX compatible operating systens. It only requires a basic C library with a BSD-style sockets API and either **fork**(2) or **pthreads**(7). That allows it to run on most embedded systems like routers, NASes, mobile phones, tablets, TVs, settop boxes, etc. Some efforts have been made that it also runs on Windows.

Although **vlmcsd** does neither require an activation key nor a payment to anyone, it is not meant to run illegal copies of Windows. Its purpose is to ensure that owners of legal copies can use their software without restrictions, e.g. if you buy a new computer or motherboard and your key will be refused activation from Microsoft servers due to hardware changes.

vlmcsd may be started via an internet superserver like **inetd**(8) or **xinetd**(8) as well as an advanced init system like **systemd**(8) or **launchd**(8) using socket based activation. If **vlmcsd** detects that **stdin**(3) is a socket, it assumes that there is already a connected client on stdin that wants to be activated. All options that control setting up listening sockets will be ignored when in inetd mode.

OPTIONS

Since vlmcsd can be configured at compile time, some options may not be available on your system.

All options that do no require an argument may be combined with a single dash, for instance "vlmcsd -D -e" is identical to "vlmcsd -De". For all options that require an argument a space between the option and the option argument is optional. Thus "vlmcsd -r 2" and "vlmcsd -r2" are identical too.

-h or **-?** Displays help.

-V Displays extended version information. This includes the compiler used to build vlmcsd, the intended platform and flags (compile time options) to build vlmcsd. If you have the source code of vlmcsd, you can type **make help** (or **gmake help** on systems that do not use the GNU version of **make**(1) by default) to see the meaning of those flags.

-L *ipaddress*[:*port*]

Instructs vlmcsd to listen on *ipaddress* with optional *port* (default 1688). You can use this option more than once. If you do not specify **-L** at least once, IP addresses 0.0.0.0 (IPv4) and :: (IPv6) are used. If the IP address contains colons (IPv6) you must enclose the IP address in brackets if you specify the optional port, e.g. [2001:db8::dead:beef]:1688.

If no port is specified, vlmcsd uses the default port according to a preceding **-P** option. If you specify a port, it can be a number (1-65535) or a name (usually found in /etc/services if not provided via LDAP, NIS+ or another name service).

If you specify a link local IPv6 address (fe80::/10, usually starting with fe80::), it must be followed by a percent sign (%) and a scope id (=network interface name or number) on most unixoid OSses including Linux, Android, MacOS X and iOS, e.g. fe80::1234:56ff:fe78:9abc**%eth0** or [fe80::1234:56ff:fe78:9abc**%2**]:1688. Windows (including cygwin) does not require a scope id unless the same link local address is used on more than one network interface. Windows does not accept a name and the scope id must be a number.

-o *level* Sets the *level* of protection against activations from public IP addresses. The default is **-o0** for no protection.

-o1 causes vlmcsd not to listen on all IP addresses but on private IP addresses only. IPv4 addresses in the 100.64.0.0/10 range (see RFC6598) are not treated as private since they can be reached from other users of your ISP. Private IPv4 addresses are 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 169.254.0.0/16 and 127.0.0.0/8. vlmcsd treats all IPv6 addresses not within 2000::/3 as private addresses.

If **-o1** is combined with **-L**, it will listen on all private IP addresses plus the ones specified by one or more **-L** statements. If **-o1** is combined with **-P**, only the last **-P** statement will be used.

Using **-o1** does not protect you if you enable NAT port forwarding on your router to your vlmcsd machine. It is identical to using multiple -L statements with all of your private IP addresses. What **-o1** does for you, is automatically enumerating your private IP addresses.

-o2 does not affect the interfaces, vlmcsd is listening on. When a clients connects, vlmcsd immediately drops the connection if the client has a public IP address. Unlike **-o1** clients will be able to establish a TCP connection but it will be closed without a single byte sent over the connection. This protects against clients with public IP addresses even if NAT port forwarding is used. While **-o2** offers a higher level of protection than **-o1**, the client sees that the KMS TCP port (1688 by default) is actually accepting connections.

If vlmcsd is compiled to use MS RPC, **-o2** can only offer very poor protection. Control is passed from MS RPC to vlmcsd after the KMS protocol has already been negotiated. Thus a client can always verify that the KMS protocol is available even though it receives an RPC_S_ACCESS_DENIED error message. vlmcsd will issue a warning if **-o2** is used with MS RPC. **For adaequate protection do not use a MS RPC build of vlmcsd with -o2**.

-o3 combines **-o1** and **-o2**. vlmcsd listens on private interfaces only and if a public client manages to connect anyway due to NAT port forwarding, it will be immediately dropped.

If you use any form of TCP level port forwarding (e.g. **nc**(1), **netcat**(1), **ssh**(1) port forwarding or similar) to redirect KMS requests to vlmcsd, there will be no protection even if you use **-o2** or **-o3**. This is due to the simple fact that vlmcsd sees the IP address of the redirector and not the IP address of the client.

-o1 (and thus **-o3**) is not (yet) available in some scenarios:

FreeBSD: There is a longtime unfixed `bug` ⟨`https://bugs.freebsd.org/bugzilla/show_bug.cgi?id=178881`⟩ in the 32-bit ABI of the 64-bit kernel. If you have a 64-bit FreeBSD kernel, you must run the 64-bit version of vlmcsd if you use **-o1** or **-o3**. The 32-bit version causes undefined behavior up to crashing vlmcsd. Other BSDs (NetBSD, OpenBSD, Dragonfly and Mac OS X) work correctly.

If vlmcsd was started by an internet superserver or was compiled to use Microsoft RPC (Windows only) or simple sockets, **-o1** and **-o3** are not available by design.

-P *port* Use TCP *port* for all subsequent **-L** statements that do not include an optional port. If you use **-P** and **-L**, **-P** must be specified before **-L**.

-F0 and **-F1**
Allow (**-F1**) or disallow (**-F0**) binding to IP addresses that are currently not configured on your system. The default is **-F0**. **-F1** allows you to bind to an IP address that may be configured after

you started **vlmcsd**. **vlmcsd** will listen on that address as soon as it becomes available. This feature is only available under Linux (IPv4 and IPv6) and FreeBSD (IPv4 only). FreeBSD allows this feature only for the root user (more correctly: processes that have the PRIV_NETINET_BINDANY privilege). Linux does not require a capability for this.

-t *seconds*

Timeout the TCP connection with the client after *seconds* seconds. After sending an activation request. RPC keeps the TCP connection for a while. The default is 30 seconds. You may specify a shorter period to free ressources on your device faster. This is useful for devices with limited main memory or if you used **-m** to limit the concurrent clients that may request activation. Microsoft RPC clients disconnect after 30 seconds by default. Setting *seconds* to a greater value does not make much sense.

-m *concurrent-clients*

Limit the number of clients that will be handled concurrently. This is useful for devices with limited ressources or if you are experiencing DoS attacks that spawn thousands of threads or forked processes. If additional clients connect to vlmcsd, they need to wait until another client disconnects. If you set *concurrent-clients* to a small value (<10), you should also select a reasonable timeout of 2 or 3 seconds with **-t**. The default is no limit.

-d Disconnect each client after processing one activation request. This is a direct violation of DCE RPC but may help if you receive malicous fake RPC requests that block your threads or forked processes. Some other KMS emulators (e.g. py-kms) behave this way.

-k Do not disconnect clients after processing an activation request. This selects the default behavior. **-k** is useful only if you used an ini file (see **vlmcsd.ini**(5) and **-i**). If the ini file contains the line "DisconnectClientsImmediately = true", you can use this switch to restore the default behavior.

-N0 and **-N1**

Disables (**-N0**) or enables (**-N1**) the use of the NDR64 transfer syntax in the RPC protocol. Unlike Microsoft vlmcsd supports NDR64 on 32-bit operating systems. Microsoft introduced NDR64 in Windows Vista but their KMS servers started using it with Windows 8. Thus if you choose random ePIDs, vlmcsd will select ePIDs with build numbers 9200 and 9600 if you enable NDR64 and build numbers 6002 and 7601 if you disable NDR64. The default is to enable NDR64.

-B0 and **-B1**

Disables (**-B0**) or enables (**-B1**) bind time feature negotiation (BTFN) in the RPC protocol. All Windows operating systems starting with Vista support BTFN and try to negotiate it when initiating an RPC connection. Thus consider turning it off as a debug / troubleshooting feature only. Some older firewalls that selectively block or redirect RPC traffic may get confused when they detect NDR64 or BTFN.

-l *filename*

Use *filename* as a log file. The log file records all activations with IP address, Windows workstation name (no reverse DNS lookup), activated product, KMS protocol, time and date. If you do not specify a log file, no log is created. For a live view of the log file type tail -f *file*.

If you use the special *filename* "syslog", vlmcsd uses **syslog**(3) for logging. If your system has no syslog service (/dev/log) installed, logging output will go to /dev/console. Syslog logging is not available in the native Windows version. The Cygwin version does support syslog logging.

-T0 and **-T1**
: Disable (**-T0**) or enable (**-T1**) the inclusion of date and time in each line of the log. The default is **-T1**. **-T0** is useful if you log to **stdout**(3) which is redirected to another logging mechanism that already includes date and time in its output, for instance **systemd-journald**(8). If you log to **syslog**(3), **-T1** is ignored and date and time will never be included in the output sent to **syslog**(3).

-D
: Normally vlmcsd daemonizes and runs in background (except the native Windows version). If **-D** is specified, vlmcsd does not daemonize and runs in foreground. This is useful for testing and allows you to simply press <Ctrl-C> to exit vlmcsd.

  The native Windows version never daemonizes and always behaves as if **-D** had been specified. You may want to install vlmcsd as a service instead. See **-s**.

-e
: If specified, vlmcsd ignores **-l** and writes all logging output to **stdout**(3). This is mainly useful for testing and debugging and often combined with **-D**.

-v
: Use verbose logging. Logs every parameter of the base request and the base response. It also logs the HWID of the KMS server if KMS protocol version 6 is used. This option is mainly for debugging purposes. It only has an effect if some form of logging is used. Thus **-v** does not make sense if not used with **-l**, **-e** or **-f**.

-q
: Do not use verbose logging. This is actually the default behavior. It only makes sense if you use vlmcsd with an ini file (see **-i** and **vlmcsd.ini**(5)). If the ini file contains the line "LogVerbose = true" you can use **-q** to restore the default behavior.

-p *filename*
: Create pid file *filename*. This has nothing to do with KMS ePIDs. A pid file is a file where vlmcsd writes its own process id. This is used by standard init scripts (typically found in /etc/init.d). The default is not to write a pid file.

-u *user* and **-g** *group*
: Causes vlmcsd to run in the specified *user* and *group* security context. The main purpose for this is to drop root privileges after it has been started from the root account. To use this feature from cygwin you must run cyglsa-config and the account from which vlmcsd is started must have the rights "Act as part of the operating system" and "Replace a process level token". The native Windows version does not support these options.

  The actual security context switch is performed after the TCP sockets have been created. This allows you to use privileged ports (< 1024) when you start vlmcsd from the root account.

  However if you use an ini, pid or log file, you must ensure that the unprivileged user has access to these files. You can always log to **syslog**(3) from an unprivileged account on most platforms (see **-l**).

-w *ePID*
: Use *ePID* as Windows ePID. If specified, **-r** is disregarded for Windows.

-0 *ePID*
: Use *ePID* as Office 2010 ePID (including Project and Visio). If specified, **-r** is disregarded for Office 2010.

-3 *ePID*

Use *ePID* as Office 2013 ePID (including Project and Visio). If specified, **-r** is disregarded for Office 2013.

-6 *ePID*

Use *ePID* as Office 2016 ePID (including Project and Visio). If specified, **-r** is disregarded for Office 2016.

-H *HwId*

Use *HwId* for all products. All HWIDs in the ini file (see **-i**) will not be used. In an ini file you can specify a seperate HWID for each *application-guid*. This is not possible when entering a HWID from the command line.

HwId must be specified as 16 hex digits that are interpreted as a series of 8 bytes (big endian). Any character that is not a hex digit will be ignored. This is for better readability. The following commands are identical:

vlmcsd -H 0123456789ABCDEF
vlmcsd -H 01:23:45:67:89:ab:cd:ef
vlmcsd -H "01 23 45 67 89 AB CD EF"

-i *filename*

Use configuration file (aka ini file) *filename*. Most configuration parameters can be set either via the command line or an ini file. The command line always has precedence over configuration items in the ini file. See **vlmcsd.ini**(5) for the format of the configuration file.

If vlmcsd has been compiled to use a default configuration file (often /etc/vlmcsd.ini), you may use **-i-** to ignore the default configuration file.

-r0, **-r1** (default) and **-r2**

These options determine how ePIDs are generated if

- you did not sprecify an ePID in the command line and
- you haven't used **-i** or
- the file specified by **-i** cannot be opened or
- the file specified by **-i** does not contain an ePID for the KMS request

-r0 means there are no random ePIDs. vlmcsd simply issues default ePIDs that are built into the binary at compile time. **Pro:** behaves like real KMS server that also always issues the same ePID. **Con:** Microsoft may start blacklisting again and the default ePID may not work any longer.

-r1 instructs vlmcsd to generate random ePIDs when the program starts or receives a SIGHUP signal and uses these ePIDs until it is stopped or receives another SIGHUP. Most other KMS emulators generate a new ePID on every KMS request. This is easily detectable. Microsoft could just modify sppsvc.exe in a way that it always sends two identical KMS requests in two RPC requests but over the same TCP connection. If both KMS responses contain the different ePIDs, the KMS server is not genuine. **-r1** is the default mode. **-r1** also ensures that all three ePIDs (Windows, Office 2010 and Office 2013) use the same OS build number and LCID (language id).

If vlmcsd has been started by an internet superserver, **-r1** works almost identically to **-r2**. The only exception occurs if you send more than one activation request over the same TCP connection. This is simply due to the fact that vlmcsd is started upon a connection request and does not stay in memory after servicing a KMS request. Consider using **-r0** or **-w**, **-0**, **-3** and **-6 when starting**

vlmcsd by an internet superserver.

-r2 behaves like most other KMS server emulators with random support and generates a new random ePID on every request. **-r2** should be treated as debugging option only because it allows very easy emulator detection.

-C *LCID*

Do not randomize the locale id part of the ePID and use *LCID* instead. The *LCID* must be specified as a decimal number, e.g. 1049 for "Russian - Russia". This option has no effect if the ePID is not randomized at all, e.g. if it is selected from the command line or an ini file.

By default vlmcsd generates a valid locale id that is recognized by .NET Framework 4.0. This may lead to a locale id which is unlikely to occur in your country, for instance 2155 for "Quecha - Ecuador". You may want to select the locale id of your country instead. See `MSDN ⟨http://msdn.microsoft.com/en-us/goglobal/bb964664.aspx⟩` for a list of valid *LCID*s. Please note that some of them are not recognized by .NET Framework 4.0.

Most other KMS emulators use a fixed *LCID* of 1033 (English - US). To achive the same behavior in vlmcsd use **-C 1033**.

-R *renewal-interval*

Instructs clients to renew activation every *renewal-interval*. The *renewal-interval* is a number optionally immediately followed by a letter indicating the unit. Valid unit letters are s (seconds), m (minutes), h (hours), d (days) and w (weeks). If you do not specify a letter, minutes is assumed.

-R3d for instance instructs clients to renew activation every 3 days. The default *renewal-interval* is 10080 (identical to 7d and 1w).

Due to poor implementation of Microsofts KMS Client it cannot be guaranteed that activation is renewed on time as specfied by the -R option. Don’t care about that. Renewal will happen well before your activation expires (usually 180 days).

Even though you can specify seconds, the granularity of this option is 1 minute. Seconds are rounded down to the next multiple of 60.

-A *activation-interval*

Instructs clients to retry activation every *activation-interval* if it was unsuccessful, e.g. because it could not reach the server. The default is 120 (identical to 2h). *activation-interval* follows the same syntax as *renewal-interval* in the **-R** option.

-s Installs vlmcsd as a Windows service. This option only works with the native Windows version and Cygwin. Combine **-s** with other command line options. These will be in effect when you start the service. The service automatically starts when you reboot your machine. To start it manually, type "net start vlmcsd".

If you use Cygwin, you must include your Cygwin system DLL directory (usually C:\Cygwin\bin or C:\Cygwin64\bin) into the PATH environment variable or the service will not start.

You can reinstall the service anytime using vlmcsd -s again, e.g. with a different command line. If the service is running, it will be restarted with the new command line.

When using **-s** the command line is checked for basic syntax errors only. For example "vlmcsd -s -L 1.2.3.4" reports no error but the service will not start if 1.2.3.4 is not an IP address on your

system.

-S Uninstalls the vlmcsd service. Works only with the native Windows version and Cygwin. All other options will be ignored if you include -S in the command line.

-U [*domain*\]*username*
Can only be used together with **-s**. Starts the service as a different user than the local SYSTEM account. This is used to run the service under an account with low privileges. If you omit the domain, an account from the local computer will be used.

You may use "NT AUTHORITY\NetworkService". This is a pseudo user with low privileges. You may also use "NT AUTHORITY\LocalService" which has more privileges but these are of no use for running vlmcsd.

Make sure that the user you specify has at least execute permission for your executable. "NT AUTHORITY\NetworkService" normally has no permission to run binaries from your home directory.

For your convenience you can use the special username "/l" as a shortcut for "NT AUTHORITY\LocalService" and "/n" for "NT AUTHORITY\NetworkService". "vlmcsd −s −U /n" installs the service to run as "NT AUTHORITY\NetworkService".

-W *password*
Can only be used together with **-s**. Specifies a *password* for the corresponding username you use with -U. SYSTEM, "NT AUTHORITY\NetworkService", "NT AUTHORITY\LocalService" do not require a password.

If you specify a user with even lower privileges than "NT AUTHORITY\NetworkService", you must specify its password. You also have to grant the "Log on as a service" right to that user.

SIGNALS

The following signals differ from the default behavior:

SIGTERM, **SIGINT**
These signals cause vlmcsd to exit gracefully. All global semaphores and shared memory pages will be released, the pid file will be unlinked (deleted) and a shutdown message will be logged.

SIGHUP
Causes vlmcsd to be restarted completely. This is useful if you started vlmcsd with an ini file. You can modify the ini file while vlmcsd is running and then sending **SIGHUP**, e.g. by typing "killall -SIGHUP vlmcsd" or "kill -SIGHUP ‘cat /var/run/vlmcsd.pid‘".

The SIGHUP handler has been implemented relatively simple. It is virtually the same as stopping vlmcsd and starting it again immediately with the following exceptions:

— The new process does not get a new process id.

— If you used a pid file, it is not deleted and recreated because the process id stays the same.

— If you used the ’user’ and/or ’group’ directive in an ini file these are ignored. This is because once you switched to lower privileged users and groups, there is no way back. Anything else would be a severe security flaw in the OS.

Signaling is not available in the native Windows version and in the Cygwin version when it runs as Windows service.

SUPPORTED OPERATING SYSTEMS

vlmcsd compiles and runs on Linux, Windows (no Cygwin required but explicitly supported), Mac OS X, FreeBSD, NetBSD, OpenBSD, Dragonfly BSD, Minix, Solaris, OpenIndiana, Android and iOS. Other POSIX or unixoid OSses may work with unmodified sources or may require minor porting efforts.

SUPPORTED PRODUCTS

vlmcsd can answer activation requests for the following products: Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10 (up to 1607), Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Office 2010, Project 2010, Visio 2010, Office 2013, Project 2013, Visio 2013, Office 2016, Project 2016, Visio 2016. Newer version may work as long as the KMS protocol does not change. A complete list of fully supported products can be obtained using the **-x** option of **vlmcs**(1).

Office, Project and Visio must be volume license versions.

FILES

vlmcsd.ini(5)

EXAMPLES

vlmcsd -De

Starts **vlmcsd** in foreground. Useful if you use it for the first time and want to see what's happening when a client requests activation.

vlmcsd -l /var/log/vlmcsd.log

Starts **vlmcsd** as a daemon and logs everything to /var/log/vlmcsd.log.

vlmcsd -L 192.168.1.17

Starts **vlmcsd** as a daemon and listens on IP address 192.168.1.17 only. This is useful for routers that have a public and a private IP address to prevent your KMS server from becoming public.

vlmcsd -s -U /n -l C:\logs\vlmcsd.log

Installs **vlmcsd** as a Windows service with low privileges and logs everything to C:\logs\vlmcsd.log when the service is started with "net start vlmcsd".

BUGS

An ePID specified in an ini file must not contain spaces.

INTENTIONAL BUGS

vlmcsd activates non-VL (retail) and beta/preview versions of Windows.
vlmcsd always reports enough active clients to satisfy the N count policy of the request.

AUTHOR

Written by crony12, Hotbird64 and vityan666. With contributions from DougQaid.

CREDITS

Thanks to CODYQX4, deagles, eIcn, mikmik38, nosferati87, qad, Ratiborus, ...

SEE ALSO

vlmcsd.ini(5), **vlmcsd**(7), **vlmcs**(1), **vlmcsdmulti**(1)