NAME

vlmcs − a client for testing and/or charging KMS servers

SYNOPSIS

vlmcs [*options*] [*target*] [*options*]

target can be one of the following:

hostname|*ipaddress*[:*tcp-port*] to query a specific KMS server (example: vlmcs kms.example.com:1688).
.domain to automatically detect KMS servers via DNS for *domain* (example: vlmcs .example.com). Please note the dot before *domain*.
- (a single dash) to detect KMS servers in your own domain.

If you use *ipaddress*:*port* as the *target*, the *ipaddress* must be enclosed in brackets if it contains colons, e.g. [2001:db8:dead:beef::1]:1688. If you use a link-local IPv6 address on Unix systems, you must append a percent sign and the interface identifier of the source interface, for example fe80::dead:beef%eth0.

If you omit the *target*, 127.0.0.1:1688 will be used except if you use **-i6**. In this case the default target is [::1]:1688.

DESCRIPTION

vlmcs is a program that can be used to test a KMS server that provides activation for several Microsoft products. The KMS server may also be an emulator. It supports KMS protocol versions 4, 5 and 6.

vlmcs generates one or more activation requests for a Microsoft KMS product and sends it to a KMS server. It then analyzes and displays the responses of the KMS server.

vlcms checks both the DCE-RPC protocol and the activation message for correctness and reports any errors that it finds.

vlmcs can also be used to "charge" a KMS server. A Microsoft KMS server sends correct activation messages only if it detects a certain minimum of clients (25 for Windows client OSses, 5 otherwise) on the network. This is Microsoft's futile attempt to prevent running a KMS server in a home environment.

OPTIONS

-h or **-?** Show help.

-V Displays extended version information. This includes the compiler used to build vlmcs, the intended platform and flags (compile time options) to build vlmcs. 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.

-x Show valid *application*s that can be used with **-l**.

-e Show some examples how to use vlmcs correctly.

-v Be verbose. Instead of just displaying the returned ePID and the HwId (protocol v6 only) vlmcsd shows all details of the query and the response.

-l *application*
Request activation for a specific *application*. Valid applications can be displayed by using **-x**. The default *application* is *Windows Vista Business*. The list of available applications is not complete. You may supply GUIDs with **-a**, **-k** and **-s** to specify applications that are not listed with **-x**. The **-l** option is used as a shortcut for the most common applications.

-K *protocol-version*

Force a specific version of the KMS protocol. Valid versions are 4.0, 5.0 and 6.0. The default is to select a suitable version according to the *application* selected. You may use **-K** to send an incorrect protocol version to the KMS server and see how it behaves. Genuine KMS servers return HRESULT 0x8007000D if the KMS protocol is not 4.0, 5.0 or 6.0. Emulators should do the same. When sending a request with an incorrect protocol number, vlmcs ignores the minor protocol number (e.g. sends a v4 request for version 4.1). If the major version number is less then 4, it sends a v4 request. If the major version is greater then 6, is sends a v6 request. In any case the *protocol-version* as specified by **-K** is put in the version fields of the request.

-4, **-5** and **-6**

Force version 4, 5 or 6 of the KMS protocol. These options are actually shortcuts of **-K 4.0**, **-K 5.0** and **-K 6.0**.

-m Let the client pretend to be a virtual machine. Early versions of Microsoft's KMS server did not increase the client count if the request came from a virtual machine. Newer versions ignore this flag.

-d Use NetBIOS names instead of DNS names. By default vlmcsd generates some random DNS names for each request. If you prefer NetBIOS names, you may use **-d**. A real Microsoft activation client uses DNS names or NetBIOS depending on the client name configuration. KMS servers treat the workstation name as a comment that affects logging only. Clients will be identified by a GUID that can be specified using **-c**. **-d** has no effect if you also specify **-w**.

-a *application-guid*

Send requests with a specific *application-guid*. There are currently only three known valid *application-guid*s:

55c92734-d682-4d71-983e-d6ec3f16059f (Windows)
59a52881-a989-479d-af46-f275c6370663 (Office 2010)
0ff1ce15-a989-479d-af46-f275c6370663 (Office 2013)

A Microsoft KMS server uses these GUIDs to have seperate counters for the already activated clients. A client that does not contact the KMS server within 30 days will be deleted from the database. Emulated KMS servers are always fully charged.

-k *kms-guid*

Send requests with a specific *kms-guid*. A Microsoft KMS server uses these GUIDs as a product id to decide whether to grant activation or not. A list of current *kms-guid*s can be found in kms.c (table KmsIdList). Emulated KMS servers grant activation unconditionally and do not check the *kms-guid*.

-s *activation-guid*

The *activation-guid* defines the actual product, e.g. "Windows 8.1 Professional WMC KMSCLIENT edition". A *activation-guid* maps 1:1 to a product key. However, neither a Microsoft KMS server nor emulated servers check this id. The *activation-guid* is useful in logging to get a specific product description like "Windows 8.1 Professional WMC". A list of current *activation-guid*s can be found in kms.c (table ExtendedProductList).

-n *requests*
Send *requests* requests to the server. The default is to send at least one request and enough subsequent requests that the server is fully charged afterwards for the *application−guid* you selected (explicitly with **-a** or implicitly by using **-l**).

-T Causes to use a new TCP connection for each request if multiple requests are sent with vlmcsd. This is useful when you want to test an emulated KMS server whether it suffers from memory leaks. To test for memory leaks use **-n** with a large number of requests (> 100000) and then test twice (with and without **-T**). This option may become neccessary for future versions of Microsoft's KMS server because multiple requests with different *clients-guid*s for the same *kms-id-guid* are impossible in a real KMS szenario over the same TCP connection.

-c *client-machine-guid*
Normally vlmcs generates a random *client-machine-guid* for each request. By using this option you can specify a fixed *client-machine-guid* This causes a Microsoft KMS not to increment its client count because it receives multiple requests for the same client. Thus do not use **-c** if you want to charge a real KMS server.

-o *previous-client-machine-guid*
If the *client-machine-guid* changes for some reason, the real KMS client stores a *previous-client-machine-guid* which is sent to the KMS server. This happens rarely and usually 00000000-0000-0000-0000-000000000000 is used. You can use **-o** to specify a different *previous-client-machine-guid*.

-G *filename*
Grabs ePIDs and HWIDs from a KMS server and writes the information to *filename* in format suitable to be used as a configuration file (aka ini file) for **vlmcsd**(8). This is especially useful if you have access to a genuine KMS server and want to use the same data with **vlmcsd**(8).

If *filename* does not exist, it will be created. If you specify an existing *filename*, it will be updated to use the information received from the remote KMS server and a backup *filename*~ will be created.

-G cannot be used with **-l**, **-4**, **-5**, **-6**, **-a**, **-s**, **-k**, **-r** and **-n**

-w *workstation-name*
Send requests with a specific *workstation-name*. This disables the random generator for the workstation name. Since it is a comment only, this option does not have much effect.

-r *required-client-count*
Also known as the "N count policy". Tells the KMS server that successful activation requires *required-client-count* clients. The default is the *required-client-count* that the product would need if the request was a real activation. A Microsoft KMS server counts clients up to the double amount what was specified with **-r**. This option can be used to "overcharge" a Microsoft KMS server.

−t *status*
Reports a specific license status to the KMS server. *status* is a number that can be from 0 to 6. 0=unlicensed, 1=licensed, 2=OOB grace, 3=OOT grace, 4=Non-genuinue grace, 5=notification, 6=extended grace. Refer to `TechNet ⟨http://technet.microsoft.com/en-us/library/ff686879.aspx#_Toc257201371⟩` for more information. A Microsoft KMS server collects this information for statistics only.

-g *binding-expiration*
This tells the KMS server how long a client will stay in its current license status. This can be the remaining OOB time (the grace peroid that is granted between installation of a product and when activation is actuall required) or the remaining time when KMS activation must be renewed. *binding-expiration* is specified in minutes. A Microsoft KMS server apparantly does not use this information.

-i *protocol-version*
Force the use of Internet protocol *protocol-version*. Allowed values are 4 (IPv4) and 6 (IPv6). This option is useful only if you specfiy a *hostname* and not an *ip-address* on the command line.

-p Do not set the RPC_PF_MULTIPLEX flag in the RPC bind request. This can be used to test if the KMS server uses the same setting of this flag in the RPC bind respone. Some KMS emulators don't set this correctly.

-N0 and **-N1**
Disables (**-N0**) or enables (**-N1**) the NDR64 transfer syntax in the RPC protocol. Disable NDR64 only in case of problems. If NDR64 is not used, vlmcs cannot detect many RPC protocol errors in KMS emulators. If you want to test whether a KMS emulator fully supports NDR64, you must use the **-n** option to send at least two requests. This is because Microsoft's client always sends the first request using NDR32 syntax and subsequent requests using NDR64 syntax.

-B0 and **-B1**
Disables (**-B0**) or enables (**-B1**) bind time feature negotiation (BTFN) in the RPC protocol. Disable BTFN only in case of problems. If BTFN is not used, vlmcs cannot detect many RPC protocol errors in KMS emulators.

Options that do not require an argument can be specified together with a single dash, e.g. vlmcs -6mvT. If you specify an option more than once, the last occurence will be in effect.

FILES

vlmcsd.ini(5)

EXAMPLES

vlmcs kms.example.com
Request activation for Windows Vista using v4 protocol from kms.example.com. Repeat activation requests until server is charged for all Windows products.

vlmcs - Request activation for Windows Vista using v4 protocol from a KMS server that is published via DNS for the current domain.

vlmcs .example.com
Request activation for Windows Vista using v4 protocol from a KMS server that is published via DNS for domain example.com.

vlmcs -6 -l Office2013 -v -n 1
Request exactly one activation for Office2013 using v6 protocol from localhost. Display verbose results.

vlmcs kms.bigcompany.com -G /etc/vlmcsd.ini
Get ePIDs and HWIDs from kms.bigcompany.com and create/update /etc/vlmcsd.ini accordingly.

BUGS

Some platforms (e.g. Solaris) may have a **man**(7) system that does not handle URLs. URLs may be omitted in the documentation on those platforms. Cygwin, Linux, FreeBSD and Mac OS X are known to work correctly.

AUTHOR

Written by Hotbird64

CREDITS

Thanks to CODYQX4, crony12, deagles, DougQaid, eIcn, mikmik38, nosferati87, qad, Ratiborus, vityan666, ...

SEE ALSO

vlmcsd(7), **vlmcsd**(8), **vlmcsdmulti**(1)