vmssc - The CLI to the Security Server

1. Overview

VMSSC is the VorMetric Securit Server Cli. In other words, it's a command line interface to the security server. It performs many functions that are available in the web-based interface today. Having vmssc is essential for automated deployments, scripting of routine/repetitive tasks, and ease of use.

This version reflects the very latest abilities of vmssc. Previous versions of this document are

- vmssc 4.4.0 Reference
- vmssc 4.4.1 Reference

2. Table of Contents

3. Implementation

The 4.4 Security Server provides a WSDL for operations regarding status, hosts, host groups, domains, administrators, etc. The CLI essentially takes some set of arguments from the user, parses them, performs the appropriate SOAP calls to the server, and displays the output. The CLI is written in C. C is used so that has zero dependencies. It will be built alongside the agent code, and can be built for each platform we support. Communications take place over port 8445, just like the web browser access to the server.

4. General Command-Line Structure

The command structure of vmssc is somewhat inspired by the way that the 'cvs' command operates. Specifically, the cvs command structure looks like

```
Usage: cvs [cvs-options] command [command-options-and-arguments]
```

There are some global options that go before the "command", and then command-specific options that go after the command.

The general format of all vmssc commands looks like the following:

```
vmssc [auth options] command_group command <command-options-and-arguments>
```

- The **[auth options]** are the same for every command_group and command. They exist so that the authentication information (like username, password, and domain) can be entered on a per-command basis. (This information can also be cached in a file; see below).
- The **command_group** specifies the top-level category to operate upon. Currently the types are "host", "group", "server", "policy", "key", "domain", and "user".
- The **command** represents some set of commands that apply to the command_group. Typical commands are "add", "delete", "modify", and "show".
- **command-options-and-arguments** are specified for each command.

While we describe the following commands as the command_group followed by the command (ex: `./vmssc host show`), it's actually also possible to reverse their order (ex: `./vmssc show host` is also legal).

4.1. Getting Help

Help is provided as part of `vmssc`. Simply executing `vmssc` with no arguments produces

```
Usage: vmssc [auth_options] <object> <command> [cmd_options] [target]
  where auth_options are -c, -d, etc.
    specify --help-auth-options for a list of options.
  where object is host, group, etc.
    specify --help-objects for a list of objects.
  where command is add, del, etc.
    specify command 'help' for object-specific help.
  Specify --help to see this message.
```

Help on all parts of the command line syntax can be found here.

- `vmssc --help-auth-options` will show the authentication options to the Security Server
- `vmssc --help-objects` shows the list of command groups available
- `vmssc help <command group>` shows help for that particular command group. For example, `vmssc help host` describes the command line syntax for host operations.

4.2. Authentication Options

There are a rich set of authentication options:

Option in Config File	Option in Environment	Option on the Command Line	Description
server	VMSSC_SERVER	-s or --server	The target server

username	VMSSC_USERNAME	-u or --username	The administrator on that server
password	VMSSC_PASSWORD	-p or --password	The password for the above. This can also be entered through a echoless prompt.
domain	VMSSC_DOMAIN	-d or --domain	The domain the command should be run inside (note: case sensitive!).
defaulthostpassword	VMSSC_DEFAULTHOSTPASSWORD	(Part of host commands)	The default value of the host password (the host add command only)
cacert	VMSSC_CACERT	-c or --ca-file	Path to a server CA certificate. This option is added to verify the authenticity of the server. It should point to a file in .pem format that contains the certificate of the CA (the root of trust) of the server. If it is not provided, the server's identity will not be verified and we may be subject to man-in-the-middle attacks. (This is just like clicking through the warnings in a web browser about a questionable certificate...)
timeout	VMSSC_TIMEOUT	-t or --timeout	How long to wait for a response (seconds). Default is 10 minutes. A value of zero means infinite.

There are many ways to give the vmssc command the above authentication information. The list below is in the order that we look for information; later information supersedes prior information. Therefore, information on the command line supersedes everything else.

1. From configuration files:
    a. `~/.vmsscpass` That's the file `.vmsscpass` in the user's home directory (%APPDATA% in windows). This file can only be created by "`vmssc server login`", as described below.
    b. `~/vmssc.conf` Again, this in the user's home directory.
    c. `vmssc.conf` The same file name, but this time in the current working directory
    d. `$VMSSC_CONF` If the `VMSSC_CONF` environment variable is set, it's value is assumed to be a file name specifying a config file.
2. From the environment
3. On the command line

Configuration files all share a common format. They are made up of general attribute=value pairs. An example file might look like

```
username = <username>
password = <password>
domain = <domain>
server = <server_name>
defaulthostpassword = <hostpass>
cacert = </path/to/ca_cert.pem>
```

A file like this can be placed in `~/vmssc.conf`, `vmssc.conf`, or the value of `$VMSSC_CONF`. The `~/.vmsscpass` file follows similar rules, but can only be created by "`vmssc server login`", again described below.

Next up are environment variables. The names of the environment variables are the same as above, but prepended with VMSSC_ and in all caps. Example:

```
export VMSSC_PASSWORD = <clever-password>
```

Putting all this together, it's possible to use a mixture of files, environment, and command line options for the [authentication options]. For example, it's possible to have the following contents of `~/vmssc.conf`:

```
username = barney
domain = domain1
server = cashcow3.i.vormetric.com
defaulthostpassword = xxxyyy123
cacert = /path/to/ca_cert.pem
```

And the following in `vmssc.conf`:

```
server = foo.bar.com
```

Then in the environment

```
VMSSC_PASSWORD=Abc1234#
```

Finally on the command line

```
./vmssc -u voradmin server show
```

The algorithm for looking at these options is as follows:

1. Try to open `~/.vmsscpass` : it doesn't exist, so move on.
2. Try to open `~/vmssc.conf` : It exists, and its contents are read. The attributes are now

```
username = barney
domain = domain1
server = cashcow3.i.vormetric.com
defaulthostpassword = xxxyyy123
cacert = /path/to/ca_cert.pem
```

3. Try to open `vmssc.conf` : It exists, and its contents are read. The attributes are now

```
username = barney
domain = domain1
server = foo.bar.com
defaulthostpassword = xxxyyy123
cacert = /path/to/ca_cert.pem
```

4. Get the VMSSC_CONF environment variable: It does not exist, so move on.
5. Read environment variables. VMSSC_PASSWORD exists, so use it. The attributes are now

```
username = barney
domain = domain1
server = foo.bar.com
defaulthostpassword = xxxyyy123
cacert = /path/to/ca_cert.pem
password = Abc1234#
```

6. Parse command line options. The -u option is found and applied. The attributes are now

```
username = voradmin
domain = domain1
server = foo.bar.com
defaulthostpassword = xxxyyy123
cacert = /path/to/ca_cert.pem
password = Abc1234#
```

With such a variety of input mechanisms, it may be confusing where certain values are coming from. For troubleshooting purposes, the "~~W" (or "~~ -which") flag has been provided. When invoked, it will show the interesting attributes, their values, and where they come from. For the above example, the output is:

```
$ ./vmssc -u voradmin -W
Required parameter SERVER (foo.bar.com) obtained from the file vmssc.conf
Required parameter USERNAME (voradmin) obtained from the command line
Required parameter PASSWORD (Abc1234#) obtained from the environment
Required parameter DOMAIN (domain1) obtained from the file /home/myoder/vmssc.conf

File /home/myoder/.vmsscpass does not exist
VMSSC_CONF is not set in the environment
```

Using the "`vmssc server login`" command is an alternate way of providing this sort information by caching credentials on the local machine. It is described in the `server login` section below.

The final wrinkle is that if all the required parameters are available, and the password is the only missing parameter, then the password will be

prompted for, like so:

```
$ ./vmssc -s toaster -u User1 server show
Password:
```

4.3. Return Codes

Exit Value	Meaning
0	Success
1	Problem parsing the command line, vmssc.conf file, etc
2	Problem communicating with the security server
3	Bad username/password, or user does not have permission on security server
4	Other error from security server

When a failure occurs, an informative reason will always be printed to stderr.

5. Command Groups

The command groups are as follows:

Name	Description
server	Shows basic information about the server
host	Actions involving the creation/modification/destruction life cycle of hosts. Also adding and removing guard points.
group	Similar to hosts, this applies to host groups. Hosts can be entered into and removed from groups
policy	Actions involving the creation, modification, and removal of policies.
key	Actions involving the creation, modification, and removal of keys. This is used to operate on both "agent keys" (those used by our products) and "vault keys" (those that we store only).
user	Actions involving the creation and maintenance of users (also known as administrators), adding and removing them from domains, etc
domain	Actions involving the creation, modification, and removal of domains

The rest of this document is committed to describing the different command groups.

6. Server Commands

The allowed server commands are "show", "login", and "logout".

6.1. Server Login and Logout

The purpose of the "login" and "logout" commands is to store a set of credentials (really, any of the [authentication options] described above) for repeated use. In the following example, there are no configuration files present:

```
$ ./vmssc -s toaster -u User1 -p User1234! -d Space server login
$ ./vmssc server show
Server Name : toaster.i.vormetric.com
Version     : 4.4.0.0
Build Num.  : uni_348v
$ ./vmssc server logout
```

What's going on here is that the "server login" command will first verify that the servername/username/password really will get you into the server, and then create `~/.vmsscpass` and put the same information into it in an obfuscated format. Then future calls will make use of that file. All that the "server logout" command does is remove `~/.vmsscpass`.

For the curious, the `~/.vmsscpass` file looks something like this:

```
$ cat ~/.vmsscpass
SERVER_OBFUSCATED = 5F538EABF11B11B8A1D98A7FECE02A78
USERNAME_OBFUSCATED = 155AA921F2F9CE823870D99979332A8A
DOMAIN_OBFUSCATED = CB69973EBB86D34FCE272442CF988721
PASSWORD_OBFUSCATED = B9BA02CBCB10079E3C2463AAD0BBB08A
PORTIBILITY_OBFUSCATED = 471C98D9E8490AF156B0649932217C92A016222317328BF46EDD95EDDA218924
EXPIRY_OBFUSCATED = 6DD48275F9C8039FB3317C09D643CD2D
DIGEST =
b1c0ffe09383b01cf60fa0e67dfdc2cd151673e89d2ff7d84ee2cefeafd565b3bb1078b080c8cd154be041102594395d731bf700a
```

Two arguments can be given to "server login".

Short Form	Long Form	Description
-r	--redistributable	Normally, the `~/.vmsscpass` file will only work on the machine that it was created on. That is, if the file is transfered to another machine it won't work. However, if this flag is specified, the `~/.vmsscpass` file can work on any machine.
-x <expiry>	--expiry <expiry>	Normally, the `~/.vmsscpass` file is valid for one day. Using this argument can change that. The <expiry> option must be of the form nnnX, where the n means a digit and X is m (minutes), h (hours), or d (days). If X is omitted the value is assumed to be hours. Examples: 12m (12 minutes), 5h (5 hours), 17d (17 days), 6 (6 hours). A value of 0 (zero) means that there is no expiration, and the `~/.vmsscpass` file can be used indefinitely.

6.2. Server Show

```
vmssc server show [options]
```

The following options are supported:

Short Option	Long Option	Description
-d	--detailed	Display detailed server information
-f	--failover	Display failover server status
-h <server>	--hosts <server>	Display hosts assigned to server
-u	--usage	Display appliance usage statistics
-a	--all	Equivalent to -d -f -u

With no options specified, this will print out very basic server information:

```
$ ./vmssc server show
Server Name : srv.my.vormetric.com
Version     : 4.4.0.0
Build Num.  : uni_340v
```

The **-d** option shows detailed server information:

```
$ ./vmssc server show -d
Server Name : cashcow3.i.vormetric.com
Version     : 4.4.0.0
Build Num.  : uni_336v
Time        : 2010-05-03 13:58:12.301 PDT
Fingerprint : 82:A2:61:E0:AA:D3:8F:87:72:D4:F6:29:B3:0C:D3:FE:7E:62:77:FC
FS License  : Valid, Expires Wed Jun 30 2010
DB2 License : Valid, Expires Wed Jun 30 2010
IDS License : Valid, Expires Wed Jun 30 2010
Key License : Valid, Expires Wed Jun 30 2010
HA Serv[0]  : cashcow3.i.vormetric.com
HA Serv[1]  : linux32-32214.qa.com
```

The **-f** option shows failover server information, in particular the status of the replication between the failover and primary nodes:

```
$ ./vmssc server show -f
cashcow3.i.vormetric.com : HA Role              : Primary
ss1.i.vormetric.com      : HA Role              : Failover
ss1.i.vormetric.com      : HA Configured        : Yes
ss1.i.vormetric.com      : HA Registered        : Yes
ss1.i.vormetric.com      : HA Last run time     : 2010-11-08 15:45:53.623
ss1.i.vormetric.com      : HA Last sync time    : 2010-11-08 15:45:53.623
ss1.i.vormetric.com      : HA Last sync status  : Success
ss1.i.vormetric.com      : HA Last sync code    : 0
ss1.i.vormetric.com      : HA Last sync message :
```

The **-u** option shows statistics regarding the resource consumption of the physical appliance box. Information on the load average, memory, and disk consumption are reported.

```
----- Uptime Information -----
 16:10:14 up  2:12,  1 user,  load average: 0.53, 0.51, 0.54

----- Filesystem Information -----
Filesystem           1M-blocks      Used Available Use% Mounted on
/dev/sda6                 9389      2987      5925  34% /
/dev/sda9               250798      4755    233304   2% /partitions/large
/dev/sda1                  891        22       824   3% /grub
tmpfs                     1963         0      1963   0% /dev/shm
/dev/sda2                 9387      2996      5915  34% /partitions/std/2
/dev/sda8                 7505       155      6970   3% /tmp

----- Virtual Memory Information -----
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0     76 159600  68896 3156572    0    0    39   209  258  524  3  0 95  1  0
```

The "**-h <server>**" option will show the hosts assigned to the given primary or failover server.

```
$ ./vmssc server show -h cashcow3.i.vormetric.com
testhosts.vormetric.com
yli-xp.vormetric.com
```

Finally, the **-a** option shows the -d, -f, and -u output, in that order.

7. Host Commands

Valid host commands are "add", "modify", "delete", "addgp", "modgp", "delgp", and "show".

7.1. host add

```
vmssc host add [opts] hostname
```

This command adds a host to the server. The options allowed are

Argument	Long Form	Short Description	Notes
-a	--autoassign	An HA server will be auto-assigned	Default behavior (flag not specified) will leave the host assigned to the primary. This is incompatible with -s, which will make the assignment manually.
-d "Description"	--description	A description of this host.	Don't forget to quote multi-word strings for the shell.
-p port	--port	The port the host uses.	Defaults to 7024.
-g hostgroup	--group	Add the host to this hostgroup.	Synonym: --hostgroup
-h hostpass	--hostpass	Set the host password.	Mutually exclusive with -G. Synonym: --password
-G	--generate	Generate the host password automatically	This essentially means "don't use the host password mechanism" - instead, the host can be unlocked using challenge-response.
-l {(as)|(As)|(AS)}	--lock	Sets the agent lock or system lock.	• "as" Means neither agent nor system lock. A synonym is "none". • "As" Turns on the agent lock but not the system lock • "AS" Turns on both locks. A synonym is "both" • Note that "aS" is disallowed. • The default behavior is "-l none".
-e {(rc)|(Rc)|(RC)}	--enable	Sets the "registration enabled" and "communication enabled" check boxes.	• There are two forms of this. In the first form, just the flags (ex: `-e "RC"`) are specfied; this will apply only to the FS agent. In the second form, the flags are shown applied to each agent just like in the `host show` command. For example: `"fs:RC db2:Rc ids:rc key:rc"` • "rc" means check neither box - registration and communication disabled. A synonym is "none". • "Rc" means check the registration box and not communication. • "RC" means both registration and communication enabled. A synonym is "both" • Note that "rC" is disallowed. • The default is "-e both" (which translates to `"fs:RC db2:rc ids:rc key:rc"`) • Agents not specified default to `rc`. Ex: `-e "ids:RC"` becomes `"fs:rc db2:rc ids:RC key:rc"` • **NOTE**: The group communication settings will override the host communication settings set here.
-s servername	--server	Assign this host to a particular HA server.	
-H <file>	--host-settings	Sets the host's "host settings" to the contents of <file>.	If <file> is "-", read from stdin.

The only required option relates to the host password. One of these three must be set:

- -G (generate the host password automatically)
- -h (set the host password)
- VMSSC_DEFAULTHOSTPASSWORD set in the environment

Ex: Add host foo.bar.com, using all the defaults:

```
vmscc host add foo.bar.com
```

Ex: Add a host with a description, and put it in the host group named "rekey_host_group"

```
vmscc host add -d "This is an example description" -g rekey_host_group foo.bar.com
```

7.2. host modify

```
vmssc host modify [opts] hostname
```

Modify the attributes of an existing host. The word "modify" may be abbreviated "mod". All of the following options are optional, however at least one must be specified. These options are almost wholly the same as the "add" options:

Argument	Long Form	Short Description	Notes
-d "Description"	--description	A description of this host.	Don't forget to quote multi-word strings for the shell.
-p port	--port	The port the host uses.	Defaults to 7024.
-g [+/-] hostgroup	--group	Add (or remove) the host to (or from) a hostgroup.	The name of the hostgroup may be prefixed with a "+" (meaning add) or a "-" (meaning remove). If neither are specified, a "+" is implied. Synonym: --hostgroup
-h hostpass	--hostpass	Set the host password.	Synonym: --password
-G	--generate	Generate the host password automatically	This essentially means "don't use the host password mechanism" - instead, the host can be unlocked using challenge-response.
-l {(as)|(As)|(AS)}	--lock	Sets the agent lock or system lock.	• "as" Means neither agent nor system lock. A synonym is "none". • "As" Turns on the agent lock but not the system lock • "AS" Turns on both locks. A synonym is "both" • Note that "aS" is disallowed. • The default behavior is "-l none".
-e {(rc)|(Rc)|(RC)}	--enable	Sets the "registration enabled" and "communication enabled" check boxes.	• There are two forms of this. In the first form, just the flags (ex: `-e "RC"`) are specfied; this will apply only to the FS agent. In the second form, the flags are shown applied to each agent just like in the `host show` command. For example: `"fs:RC db2:Rc ids:rc key:rc"` • "rc" means check neither box - registration and communication disabled. A synonym is "none". • "Rc" means check the registration box and not communication. • "RC" means both registration and communication enabled. A synonym is "both" • Note that "rC" is disallowed. • The default is "-e both" (which translates to only modifying the FS agent to `RC` • Agents not specified aren't altered. Ex: `-e "ids:RC"` will only change the IDS agent to `RC` • **NOTE**: The group communication settings will override the host communication settings set here.
-s servername	--server	Assign this host to a particular HA server.	
-H <file>	--host-settings	Sets the host's "host settings" to the contents of <file>.	If <file> is "-", read from stdin.

Ex: Remove foo.bar.com from one hot group and then put it into another

```
vmssc host mod -g -rekey_host_group
vmssc host mod -g +production_host_group
```

7.3. host delete

```
vmssc host delete [opts] hostname
```

Delete a host. The word "delete" may be replaced with "del".

There is only one option:

- **-f** Force. Without this flag, the host is not deleteable if is not reachable by the server. When the flag is set, the server will delete it regardless.

Ex: delete foo.bar.com

```
vmssc host del -f foo.bar.com
```

7.4. host addgp

This adds a guard point to a host. Format: `vmssc host addgp [options] hostname`. Options:

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.
- **-t type** The guard point type. Valid values are "dir", "manualdir", "raw", and "manualraw". If not specified, the default is "dir".
- **-a** If specified, this is an automount guard point. (Default: not automount.)
- **-e** Enable, or **-x** Disable. Default: enabled

Example:

```
vmssc host addgp -d /full/path/to/gp -p policyname foo.bar.com
```

7.5. host modgp

This modifies a guard point on a host. Currently there is only one thing that can be modified, and that is whether the GP is enabled or not. Format: `vmssc host modgp [options] hostname`. Options

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.
- **-e** Enable, or **-x** Disable.

Example: disable the above GP

```
vmssc host modgp -d /full/path/to/gp -p policyname -x foo.bar.com
```

7.6. host delgp

Removes a guard point from a host. Format: `vmssc host delgp [options] hostname`. Options:

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.

Both the directory and policy are both required because the following scenario could exist: two guardpoints with the same path but different policies, and one is disabled and one enabled.

Example: remove the above gp

```
vmssc host delgp -d /full/path/to/gp -p policyname foo.bar.com
```

7.7. host show

```
vmssc host show [hostname]
```

Display information for a host. If given no host name, it will display all known hosts, one per line. If given a host name, it will display information for the host in the following format. We have chosen this format with the desire that is both readable by a human and parsable by a machine.

Options:

- **-H** Show the host settings in addition to all the other information.

Output format:

```
Hostname    : hostname
Description : desc
OS Info     : 'osType, ossubtype' or 'Not yet registered'
Port        : port
Locked      : (no)|(system)|(fs agent)|(system, fs agent)
Comm flags  : fs:RC db2:RC ids:RC key:RC
Version[a]  : version
Finger[a]   : fingerprint
GP[x]       : (Enabled|Disabled) policy /full/path
Group[x]    : group
```

The RC|Rc|rc in the comm flags corresponds to the -e option of add/mod above. The [a] stands for agent: fs, db2, ids. Those lines are only printed if an agent of the corresponding type is registered.

An example:

```
Description : This is a description
OS Info     : Linux, RHEL 5
Port        : 7024
Locked      : no
Comm flags  : fs:RC db2:RC ids:RC key:RC
Version[fs] : 4.3.6.0-Build558v
Finger[fs]  : DB:C7:19:0E:A1:BB:99:88:E9:34:22:1F:5F:33:7B:35:91:8F:2E:14
GP[1]       : enabled  encrypt_things /full/path/to/gp/one
GP[2]       : disabled other_policy /full/path/to/gp/two
Group[1]    : encrypting_group
Group[2]    : other_group
```

8. Group Commands

Valid group commands are "add", "modify", "delete", "addgp", "modgp", "delgp", and "show". The word "group" may be replaced with synonyms "hostgroup" and "hg".

8.1. group add

```
vmssc group add [opts] groupname
```

The options here are quite similar to their host-based counterparts:

Argument	Long Form	Short Description	Notes
-d "Description"	--description	A description of this host.	Don't forget to quote multi-word strings for the shell.

-l {(as)|(As)|(AS)}	--lock	Sets the agent lock or system lock.	• "as" Means neither agent nor system lock. A synonym is "none". • "As" Turns on the agent lock but not the system lock • "AS" Turns on both locks. A synonym is "both" • Note that "aS" is disallowed. • The default behavior is "-l none".
-e {C|c}	--enable	Sets the "communication enabled" check box.	• There are two forms of this. In the first form, just the flags (ex: `-e "C"`) are specfied; this will apply only to the FS agent. In the second form, the flags are shown applied to each agent just like in the `host show` command. For example: `"fs:C db2:c ids:c key:c"` • "C" means set "communication enabled". • "c" disables communication. • The default is "C" (which translates to `"fs:C db2:c ids:c key:c"`) • Agents not specified default to `c`. Ex: `-e "ids:C"` becomes `"fs:c db2:c ids:C key:c"` • **NOTE**: This communication setting overrides that of member hosts.
-h host	--host	Add host to this host group.	

Ex: Add a host group 'groupname' with one guard point:

```
vmssc group add -d "This is an example host group description" -e:C groupname
```

8.2. group modify

```
vmssc group modify [opts] groupname
```

The word "modify" may be replaced by "mod"
The options here are again quite similar to the above:

Argument	Long Form	Short Description	Notes
-d "Description"	--description	A description of this host.	Don't forget to quote multi-word strings for the shell.
-l {(as)|(As)|(AS)}	--lock	Sets the agent lock or system lock.	• "as" Means neither agent nor system lock. A synonym is "none". • "As" Turns on the agent lock but not the system lock • "AS" Turns on both locks. A synonym is "both" • Note that "aS" is disallowed. • The default behavior is "-l none".
-e {C|c}	--enable	Sets the "communication enabled" check box.	• There are two forms of this. In the first form, just the flags (ex: `-e "C"`) are specfied; this will apply only to the FS agent. In the second form, the flags are shown applied to each agent just like in the `host show` command. For example: `"fs:C db2:c ids:c key:c"` • "C" means set "communication enabled". • "c" disables communication. • The default is "C" (which translates to modifying the FS agent to `C`) • Agents not specified default to `c`. Ex: `-e "ids:C"` will modify just the IDS agent. • **NOTE**: This communication setting overrides that of member hosts.
-h [+/-] host	--host	Add host to this host group.	The name of the host may be prefixed with a "+" (meaning add) or a "-" (meaning remove). If neither are specified, a "+" is implied.
-p password	--password	Set all host passwords.	Sets the host password to "password" for all hosts in the host group. Note that this can take some time if there are many hosts in the target host group. Also note that this only affects hosts that are in the group at the time this command is issued; hosts added later won't be changed.

-G	--generate	Generate all host passwords	Re-generate all host passwords for all hosts in the host group. This essentially means "don't use the host password mechanism" - instead, the host can be unlocked using challenge-response.

Ex: change the description of a host group

```
vmssc group modify -d "Different description" groupname
```

8.3. group delete

```
vmssc group delete groupname
```

Deletes a host group. "delete" may be replaced with "del". This command takes no options.

Ex:

```
vmssc group del groupname
```

8.4. group addgp

This adds a guard point to a host group. Format: `vmssc group addgp [options] groupname`. Options:

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.
- **-t type** The guard point type. Valid values are "dir", "manualdir", "raw", and "manualraw". If not specified, the default is "dir".
- **-a** If specified, this is an automount guard point. (Default: not automount.)
- **-e** Enable, or **-x** Disable. Default: enabled

Example:

```
vmssc group addgp -d /full/path/to/gp -p policyname groupname
```

8.5. group modgp

This modifies a guard point in a group. Currently there is only one thing that can be modified, and that is whether the GP is enabled or not. Format: `vmssc group modgp [options] groupname`. Options

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.
- **-e** Enable, or **-x** Disable.

One of -e or -x is required. The "enabled" ability cannot be set upon GP creation (they're enabled by default), so it must be turned on/off here.

Example: disable the above GP

```
vmssc group modgp -d /full/path/to/gp -p policyname -x groupname
```

8.6. group delgp

Removes a guard point from a host group. Format: `vmssc group delgp [options] hostname`. Options:

- **-d directory** The full path of the guard point. Also could mean d for "device" for the path to a raw device. Required.
- **-p policy** The policy to apply. Required.

Both the directory and policy are both required because the following scenario could exist: two guardpoints with the same path but different policies, and one is disabled and one enabled.

Example: remove the above gp

```
vmssc group delgp -d /full/path/to/gp -p policyname groupname
```

8.7. group show

```
vmssc group show [groupname]
```

With no groupname, this command will print out all known host groups, one per line.

If a host group is specified, it will display information for the group in the following format. We have chosen this format with the desire that is both readable by a human and parsable by a machine.

```
Description : desc
Locked      : (no)|(system)|(fs agent)|(system, fs agent)
Comm flags  : (RC)|(Rc)|(rc)
GP[x]       : (enabled|disabled) policy /full/path
Member[x]   : host
```

Example:

```
Description : This is a description.
Locked      : no
Comm flags  : RC
GP[1]       : enabled  policy_foo /full/path/foo
GP[2]       : disabled policy_bar /full/path/bar
Member[1]   : alpha.beta.com
Member[2]   : gamma.beta.com
```

9. Policy Commands

The policy commands permit the creation and manipulation of policies. Policies are in XML format.

9.1. Policy Show

Displays the names of all the policies in the system. This is helpful for selecting a policy before applying it to a guard point. This command can also be used to save a policy to a filename or display it to stdout.

```
$ vmssc policy show
Dataxform
Production
UnDataxform
test1
test2
permit_apply_key_3
```

```
$ vmssc policy show -f <filename> test1
Policy Name    : test1
Policy Version : 1
Description    : (none)
```

```
$ vmssc policy show -f <filename> test1
Policy Name    : test1
Policy Version : 1
Description    : (none)
<?xml version="1.0" encoding="UTF-8" standalone="no"?><Policy Version="0">
<Target>
<Users Name="Contact">
</Users>
<Processes Name="Process">
<ProcessSet Name="process0">
</ProcessSet>
</Processes>
<Libs Name="Library">
</Libs>
<ProcessLibs/>
<Resources Name="Resource">
</Resources>
<Actions Name="Action">
</Actions>
<Times Name="Time">
</Times>
</Target>
<KeyRules KeyCombiningAlg="first_applicable">
</KeyRules>
<NewKeyRules KeyCombiningAlg="first_applicable">
</NewKeyRules>
<SecurityRules PartialMatch="1" NeverDeny="0" PermitCombiningAlg="first_applicable">
</SecurityRules>
</Policy>
```

9.2. Policy Save

Allows an administrator to save a policy from a file or stdin (think: piped in from something else). To use stdin, specify "-" as the input file.

```
$ vmssc policy save [-d "description"] [-f <filename>] policyname
```

9.3. Policy Delete

Allows an administrator to delete a policy.

```
$ vmssc policy delete policyname
```

10. Key Commands

Key commands are used for both agent keys and key vaulting.

10.1. key show

Without a name, key show will display a list of the keys available on the system. Key show can be used to display lists of keys, or one key in particular. It accepts the following flags:

Argument	Long Form	Short Description	Notes
-a	--agent	Display agent keys only	
-v	--vaulted	Display vaulted keys only	
-d	--detail	Provide details on each key	
-f	--key-file	Save the symmetric key to this file	Requires a named key

-P	--private-key	Save the private key to this file	Requires a named key
-p	--public-key	Save the public key to this file	Requires a named key

Examples:

```
$ ./vmssc show key
agent-host-1
clear_key
vaulted-aes-1
vaulted-aes-2
vaulted-aes-3
vaulted-rsa-2048
```

```
$ ./vmssc show key -a
agent-host-1
clear_key
```

```
$ ./vmssc show key -d clear_key
Key Name         : clear_key
Algorithm        : (null)
Description      : Note: Clear Key can not be deleted
Creation Time    :
Expiration Time  :
Key Type         : Stored on Server
```

10.2. key add

Allows an administrator to import or create a key.

Argument	Long Form	Scope	Short Description	Notes
-d	--description	global	Key description	
-c	--created	global	Creation date of the key.	If not set, the current date will be used. The timestamp format is yyyy-mm-dd hh:mm:ss[.fffffffff]
-e	--expires	global	Expiration date of the key.	The timestamp format is yyyy-mm-dd hh:mm:ss[.fffffffff]
-t	--type	global	Key type or algorithm	Valid values are AES128, AES256, ARIA128, ARIA256, 3DES, RSA1024, RSA2048, and RSA4096
-f	--file	global	A file containing the contents of a symmetric key.	If the file is "-", read contents from stdin.
-P	--private-key	global	A file containing the contents of an asymmetric private key	If the file is "-", read contents from stdin.
-p	--public-key	global	A file containing the contents of an asymmetric public key	If the file is "-", read contents from stdin.
-V	--create	agent	Create the key (not vaulted)	
-h	--host	agent	Agent key is cached on host	Default behavior. One of -h, -s, or -v is required.
-s	--server	agent	Agent key is stored on server (not cached on host)	One of -h, -s, or -v is required.
-u	--no-unique	agent	Key is non-unique to host	Opposite of -U. Default behavior.
-U	--unique	agent	Key is unique to host	Opposite of -u. Don't use this type of key when guarding shared storage.
-v	--valuted	vault	Vault the key	One of -h, -s, or -v is required. Vaulted keys are assumed to be third party keys.
-l	--validate	vault	Validate the key before vaulting	Default behavior.

-L	--no-validate	vault	Do not validate the key before vaulting	

Examples:

```
$ ./vmssc key add -d "My first key" -e "2015-04-01 01:00:00" -t AES256 -h first-agent-key

$ ./vmssc key show first-agent-key
Key Name         : first-agent-key
Algorithm        : AES256
Description      : My first key
Creation Time    : 2012-03-29 14:17:19.04
Expiration Time  : 2015-04-01 01:00:00.0
Key Type         : Cached on Host
```

```
$ ./vmssc key add -d "My first vaulted key" -t AES256 -f key256.txt -v -l first-vault-key

$ ./vmssc key show first-vault-key
Key Name         : first-vault-key
Algorithm        : AES256
Description      : My first vaulted key
Creation Time    : 2012-03-29 14:22:39.705
Expiration Time  : (none)
Key Type         : Vaulted
```

```
$ ./vmssc key add -d "My first RSA vaulted key" -t RSA2048 -P my.priv.pem -p my.pub.pem -v
first-rsa-vaulted

$ ./vmssc key show first-rsa-vaulted
Key Name         : first-rsa-vaulted
Algorithm        : RSA2048
Description      : My first RSA vaulted key
Creation Time    : 2012-03-29 14:42:48.592
Expiration Time  : (none)
Key Type         : Vaulted
```

10.3. key modify

Allows an administrator to modify settings for an existing key.

Argument	Long Form	Short Description	Notes
-d	--description	Key description	
-c	--created	Creation date of the key.	If not set, the current date will be used. The timestamp format is yyyy-mm-dd hh:mm:ss[.fffffffff]
-e	--expires	Expiration date of the key.	The timestamp format is yyyy-mm-dd hh:mm:ss[.fffffffff]
-h	--host	Agent key is cached on host	Default behavior. One of -h, -s, or -v is required.
-s	--server	Agent key is stored on server (not cached on host)	One of -h, -s, or -v is required.

Example:

```
$ ./vmssc key mod -d "Modified rsa key description" -e "2016-01-02 12:34:56" first-rsa-vaulted

$ ./vmssc key show first-rsa-vaulted
Key Name         : first-rsa-vaulted
Algorithm        : RSA2048
Description      : Modified rsa key description
Creation Time    : 2012-03-29 14:42:48.592
Expiration Time  : 2016-01-02 12:34:56.0
Key Type         : Vaulted
```

10.4. key setattr

Allows an administrator to set attributes for an existing key.

Argument	Long Form	Short Description	Notes
-a	--attribute	An attribute as a "name=value" pair.	Spaces are allowed in the name component
-A	--attribute-file	A file containing "name=value" pairs.	Spaces are not allowed in the name component

Example:

```
$ ./vmssc key setattr -a "Usage=Experimentation" first-rsa-vaulted

$ cat attrs.txt
Organization = Development
Purpose = Example
FavoriteColor = Blue

$ ./vmssc key setattr -A attrs.txt first-rsa-vaulted

$ ./vmssc key show first-rsa-vaulted
Key Name         : first-rsa-vaulted
Algorithm        : RSA2048
Description      : Modified rsa key description
Creation Time    : 2012-03-29 14:42:48.592
Expiration Time  : 2016-01-02 12:34:56.0
Key Type         : Vaulted
Attributes[ 0]   : FavoriteColor=Blue
Attributes[ 1]   : Organization=Development
Attributes[ 2]   : Purpose=Example
Attributes[ 3]   : Usage=Experimentation
```

10.5. key delattr

Allows an administator to remove an atribute from an existing key.

Argument	Long Form	Short Description	Notes
-a	--argument	Name of the attribute to delete	

Example:

```
$ ./vmssc key delattr -a "Purpose" first-rsa-vaulted

$ ./vmssc key show first-rsa-vaulted
Key Name         : first-rsa-vaulted
Algorithm        : RSA2048
Description      : Modified rsa key description
Creation Time    : 2012-03-29 14:42:48.592
Expiration Time  : 2016-01-02 12:34:56.0
Key Type         : Vaulted
Attributes[ 0]   : FavoriteColor=Blue
Attributes[ 1]   : Organization=Development
Attributes[ 2]   : Usage=Experimentation
```

10.6. key delete

Allows an administrator to remove an existing key.

Example:

```
$ ./vmssc key delete first-rsa-vaulted
```

11. User Commands

The user commands permit the creation of users, the assignment of users to domains, and the removal of users. The terms "user" and "administrator" are interchangeable. The user commands are a little different in that they require significant knowledge of how domains work in the Security Server and what abilities different account types have. A corresponding wrinkle is that the domain that one operates in is implicit in the [authentication options] part of the command, not in the [command arguments] part of the command. This stems from the inextricable nature of actions being performed either inside a particular domain or outside of all domains.

This table should server as a brief overview of the abilities of different user types:

Type	Abbreviation	Create User	Assign To Domain	Audit/Host/Policy/Key
System	sys	X	First only	
Domain	dom		X	
Security	sec			X
DomainSecurity	domsec or ds		X	X
All		X	X	X

The sub-commands "show", "add", "mod", and "del" do pretty much you'd expect at this point; the new sub-commands "adddom", "moddom", and "deldom" are used to add, modify, and remove a user to or from a domain.

11.1. User Show

Display the properties of a user. With no arguments, all the users in the system are displayed. If operating inside a domain, the users inside that domain are then displayed.

Example:

Outside of a domain:

```
$ ./vmssc user show
admin
domadmin
foo
glarch
```

Inside a domain:

```
$ ./vmssc -d domain1 user show
--- Administrators ---
admin
domadmin
foo
glarch
--- Members of Domain "domain1" ---
domadmin
foo
glarch
```

For a particular user:

```
$ ./vmssc user show admin
Name        : admin
Type        : System
Description : Initial system account
```

11.2. User Add

Create a user account. Arguments:

Argument	Long Form	Short Description	Notes
-d "Description"	--description	A description of this user.	Don't forget to quote multi-word strings for the shell.
-p password	--password	The initial password for this user.	The user must change this password upon first login. Required.
-t type	--type	Type of user account to create.	The types are specified in the table above. Default: Security
-D	--addtodom --add2dom --domadd	Add the newly created user to a domain.	The domain is specified in the [authentication options] section of the command, and can come from the command line, vmssc.conf file, or the environment. Note that this command can fail if the user doing the creating does not have the ability to assign additional users to a domain (as is the case with the System type).
-r roles	--role --roles	The roles to assign to the user in the domain.	Requires -D to be used. Valid roles are "audit", "host", "key", and "policy". Combinations are possible with a list separated by spaces and commas. Examples: "audit, host"; "audit, host, key, policy". Shortcuts are "all", and each role can be abbreviated to its first letter: "a,h,k,p".

Examples:

```
$ ./vmssc user add -p P@5sw0rd -d "This user is a newbie" newuser
$ ./vmssc user show newuser
Name        : newuser
Type        : Security
Description : This user is a newbie
```

```
$ ./vmssc -d domain1 user add -d "All powerful" -t all -p P@5sw0rd -D -r all master
$ ./vmssc -d domain1 user show master
Name        : master
Type        : All
Description : All powerful
Domain domain1 : Enabled : Yes
Domain domain1 : Roles   : audit, key, policy, host
```

11.3. User Modify

This command modifies attributes of an existing user.

Argument	Long Form	Short Description	Notes
-d "Description"	--description	A description of this user.	Don't forget to quote multi-word strings for the shell.
-p password	--password	The initial password for this user.	This will reset the user's password.

Example:

```
$ ./vmssc -d domain1 user add -d "All powerful" -t all -p P@5sw0rd master
$ ./vmssc -d domain1 user show master
Name        : master
Type        : All
Description : All powerful

$ ./vmssc user mod -d "Secret flaw: kryptonite" master
$ ./vmssc user show master
Name        : master
Type        : All
Description : Secret flaw: kryptonite

$ ./vmssc user mod -p NewP@5sw0rd master
```

11.4. User Del

Remove the specified user from the system. Takes one argument: the user to delete.

Example:

```
$ ./vmssc user del
Expected a username
$ ./vmssc user del master
[USR0409E] Error from cashcow3.i.vormetric.com: Users "master" cannot be deleted because they are in
domains.
$ ./vmssc -d domain1 user deldom master
$ ./vmssc user del master
$ ./vmssc show user master
[USR0160E] Error from cashcow3.i.vormetric.com: User name "master" does not exist.
```

11.5. User Adddom

Adds a user to a domain. A synonym is "domain adduser".

Argument	Long Form	Short Description	Notes
-r roles	--role --roles	The roles to assign to the user in the domain.	Valid roles are "audit", "host", "key", and "policy". Combinations are possible with a list separated by spaces and commas. Examples: "audit, host"; "audit, host, key, policy". Shortcuts are "all", and each role can be abbreviated to its first letter: "a,h,k,p".

11.6. User Moddom

Modifies the domain attributes of a user (for example, if they're enabled in a domain). A synonym is "domain moduser".

Argument	Long Form	Short Description	Notes

-r roles	--role --roles	The roles to assign to the user in the domain.	Valid roles are "audit", "host", "key", and "policy". Combinations are possible with a list separated by spaces and commas. Examples: "audit, host"; "audit, host, key, policy". Shortcuts are "all", and each role can be abbreviated to its first letter: "a,h,k,p".
-e	--enable	Enable a user in the domain	Obviously a counterpart to -x below.
-x	--disable	Disable a user in the domain	The opposite of -e.

11.7. User Deldom

Remove a user from a domain. A synonym is "domain deluser". This command takes no arguments.

Examples of use of adddom, moddom, and deldom:

```
$ ./vmssc user add -p P@sSw0rd -d "newb" newuser

$ ./vmssc -d Earth user adddom -r policy,key newuser

$ ./vmssc -d Earth user show newuser
Name        : newuser
Type        : Security
Description : newb
Domain Earth Enabled : Yes
Domain Earth Roles   : key, policy

$ ./vmssc -d Earth user moddom -r all -x newuser

$ ./vmssc -d Earth user show newuser
Name        : newuser
Type        : Security
Description : newb
Domain Earth Enabled : No
Domain Earth Roles   : audit, key, policy, host

$ ./vmssc -d Earth user deldom newuser

$ ./vmssc -d Earth user show newuser
Name        : newuser
Type        : Security
Description : newb
```

12. Domains

Administrative Domains (just "Domains" in this document) can be thought of as "silos" in which administrators, keys, policies, and hosts reside. This set of commands creates, modifies, and removes them.

12.1. Domain Show

Like the other "show" commands, if no arguments are given all the domains are shown; if the name of a domain is given then the properties of that domain are shown. Examples:

```
$ ./vmssc domain show
domain1
domain2
Earth
```

```
$ ./vmssc domain show Earth
Name        : Earth
Description : Whole world
```

12.2. Domain Add

Create a domain. Arguments

Argument	Long Form	Short Description	Notes
-d description	--description	Set the description for this domain	Optional
-u user	--user	The first administrator for this domain	Optional. Once one is specified, another cannot be added.

```
$ ./vmssc domain add -d "The Final Frontier" -u voradmin Space
```

```
$ ./vmssc domain show Space
Name        : Space
Description : The Final Frontier
```

12.3. Domain Modify

The arguments to domain modify are precisely the same as those for domain add. Example:

```
[myoder@dev vmssc]$ ./vmssc domain mod -d "These are the voyages of the Starship Enterprise" Space
```

```
$ ./vmssc show domain Space
Name        : Space
Description : These are the voyages of the Starship Enterprise
```

12.4. Domain Delete

Remove a domain. One argument accepted: the name of the domain. Example:

```
$ ./vmssc domain del Space
```