

# ScreenOS Hidden Commands Revealed

---

Version 4.01, 12/10/2002

Stephen Gill  
E-mail: [gillsr@cymru.com](mailto:gillsr@cymru.com)  
Published: 12/10/2002

---

# Contents

Revision History .....	2
Introduction .....	3
Commands .....	3
asic .....	3
cm .....	4
config .....	4
console .....	4
counter .....	4
dbuf .....	4
debug .....	5
dns .....	5
filter .....	5
flow .....	5
fragguard .....	6
ftp .....	6
h323 .....	6
interface .....	6
mac-learn-sticky .....	7
net-pak .....	7
nvram .....	7
policy .....	7
rms .....	7
session .....	8
snoop .....	8
sys-cfg .....	8
system .....	8
undebg .....	8
vpnmonitor .....	9
To Be Determined .....	9
Conclusion .....	9
References .....	10

# Credits

My thanks to those who have contributed to some of the contents of this document including

- Dave Klein [[dklein@netscreen.com](mailto:dklein@netscreen.com)]
- Jeremy Stinson [[jstinson@quadrix.com](mailto:jstinson@quadrix.com)]
- Graham Morris [[graham.morris@vanco.co.uk](mailto:graham.morris@vanco.co.uk)]

# Revision History

The document version has been updated to match the current ScreenOS version. Commands presented in this document should be current relative to the version of ScreenOS the version number corresponds to.

Version	Date	Description
1.0	09/17/2002	First unofficial draft of undocumented commands released to [nn] [1].
2.0	12/05/2002	Second revision released to the [nn] mailing list with minor updates, including a new section on Flows.
4.0	TBD	Added over 30 undocumented commands and revamped the structure and format of the document.

**Table 1 - Revision History**

## Introduction

At times I have found myself troubleshooting issues in the ScreenOS CLI only to discover that the resolution to a problem requires the entry of an undocumented command. Other times I have used troubleshooting commands that are not readily documented which have proven invaluable in problem determination.

In an effort to improve the end user's ability to troubleshoot issues on the ScreenOS CLI, I've decided to compile a list of undocumented commands in a concise format. One thing to keep in mind is that these commands are undocumented for a reason! Be sure you understand exactly what you are doing before making use of them and preferably test in a lab before using them in a production environment. For instance, pay particular care when using the 'snoop' command. I have been known to lock myself out of a device once or twice due to increased system utilization. Also keep in mind that some of these commands are only available on certain ScreenOS versions while they may be documented in others.

These 'undocumented' commands are usually (but not always) hidden for one of four reasons:

1. It is **brand new** and is still being tested for effectiveness and functionality.
2. The command is **custom made** to solve a particular customer problem that may have been brought into mainline code without notifying Tech Pubs.
3. It is a **legacy** command that remains for backward compatibility. Its use may be deprecated in favor of a newer command or syntax.
4. It is an **engineering** command that is designed for experts or internal use only.

For the purpose of this document we are mostly interested in numbers 1, 2, and 4. Deprecated commands, although historically interesting, do not add the same amount of value as other commands and will mostly be left out of this document. Commands that have a corresponding opposite such as 'set' and 'unset' will usually be listed as a single entry for brevity.

If the reader is aware of other undocumented system features, settings, or simply explanations that may fit into such a list, feel free to share them and join the credits section! The end result will hopefully be an improved ability to support and troubleshoot Netscreen firewalls.

## Commands

Instead of listing commands categorically, they have been placed alphabetically to better assist the reader in possibly finding an appropriate entry and to maintain consistency with current Netscreen CLI documentation. Additionally, most CLI variables and dependency delimiters are also maintained for consistency with Netscreen documentation.

### asic

```
get asic acl
```

Display ASIC limits comparing current use to maximum configurable ACLs.

## **cm**

```
get cm <1-4>
```

View some of the syntax associated with one of the four major command menus. The argument expected is an index of each of the top level keywords including: set, get, clear, exec. The output of this command is verbose but lists what ScreenOS expects in terms of command line arguments.

## **config**

```
get config checksum
```

Display only the global configuration checksum. It can be useful when quickly comparing configurations to see if alterations have been made.

## **console**

```
set console dbuf
```

This command is documented (strangely enough) but should be used in conjunction with commands that are verbose in output so as to not hog the console. This redirects all debug output to a buffer instead of the console.

```
set console change-notification-character <character>
```

Nice little command to enable a change notification character on the CLI. If the configuration changes, the specified character will appear on the CLI prompt until it is saved. The "+" character might be handy for this purpose.

## **counter**

```
get counter info
```

Display detailed counter information including number of counters configured, associated policy id, and time elapsed on system counters (second, minute, hour, day, month).

```
get counter ha
```

Returns information on the HA interface's hardware counters. This includes in packets, out packets, CRCs, no aligns, no buffers, collisions, underruns.

## **dbuf**

```
get dbuf <arguments>
info          show debug buffer info
mem          show debug buffer memory content
stream       show debug buffer stream
```

This allows you to view console messages that have been redirected to a debug buffer above.

```
set dbuf size <size>
```

Increase the size of the dbuf buffer from the default of 32k.

## debug

```
debug <arguments>
```

Debug is extremely handy for troubleshooting most firewall issues. It should be used in conjunction with 'set console dbuf' and 'get dbuf' commands if possible. Following are a few of the debug options that can be particularly helpful.

```
debug flow basic
```

This will show what the flow engine is doing with each packet traversing the Netscreen (e.g., packet dropped denied by policy, packet allowed by policy id X, packet being routed out interface e3, etc.).

```
debug ike detail
```

This is good for using when trying to debug ISAKMP (IKE) tunnel setups (e.g., detect mis-matched proposals, mis-matched phase 2 proxy id's [tunnel selectors], can't find gateway, etc.).

```
debug pki detail
```

This is good for debugging the use of X.509 certificates within IKE.

```
get debug
```

List the current debug flags that are enabled.

## dns

```
set dns udp-session-normal
```

Enable the normal handling of DNS UDP packets. Helpful when multiple queries are issued with the same source port so that return queries will be allowed through instead of just the first one (IE BIND).

## ffilter

```
get ffilter
```

Display the filters used for the display of debug flow output including parameters for source IP, dest IP, source port, dest port, and IP protocol. In some code versions 'set ffilter' will show up as an option but 'get ffilter' will not.

## flow

```
set flow log <arguments>
dst-ip          dst ip
dst-port        dst port
proto          ip proto
```

```
src-ip          src ip
src-port        src port
```

Restrict the flow logging information to a specific subset of traffic

```
set flow session
```

Configure the TCP session cleanup time in intervals of 10 seconds. The system default has been recently decreased to 2 seconds instead of 10 so do not use this unless you have to since the smallest time you can set is 10 seconds.

```
.get flow <arguments>
<return>          show current flow configuration settings
perf              show flow perf stats
tcp-mss           show TCP maximum segment size for VPN tunnel
```

View flow settings including timeouts, cleanup time, action flags, syn flag checking, and more.

```
set flow vpn-untrust-mip
```

Enable MIP translation for IP addresses that traverse a VPN. Use 'unset' to disable this.

## **fragguard**

```
unset fragguard
```

Refer to Netscreen id# nskb2701. If the number of fragmented packets is high, and determined NetScreen has run out of net-pak, the workaround is to run this flag.

## **ftp**

```
set ftp non-rfc-support
```

Refer to NetScreen id# nskb3258. This allows you to make passive FTP connections to servers that do not follow the RFC i.e. Cisco FTP and Microsoft FTP server. This issue has been resolved in ScreenOS 4.0.0.r5.

## **h323**

```
set h323 gate source-port-any
```

Change the system default to remove restrictions on the h323 gate source port.

```
get h323
```

Display current parameters of h323 source port restrictions.

## **interface**

```
set interface <interface> no-subnet-conflict-check
```

Disable subnet conflict checking. This allows you to configure multiple interfaces in the same IP broadcast domain!

## mac-learn-sticky

```
set mac-learn-sticky
```

Enable sticky mac learning when the firewall is in transparent mode. This will disable the automatic aging of learned MAC entries. System default is to age out old entries.

## net-pak

```
get net-pak <arguments>
<return>
distribute          net data pak distribution
link                net data pak in link
stats               net data pak statistics
```

Return information on memory pool allocations, hits, and misses based on buffer sizes from tiny to giants.

## nvrnm

```
get nvrnm
```

Display nvrnm magic number, checksum, flags, and software version.

## policy

```
get policy asic
```

Tells you how many rules you have created and what the maximum number allowable is regardless of policy direction.

```
get policy incoming asic
get policy outgoing asic
get policy fromdmz asic
get policy todmz asic
```

Commands included here for backwards compatibility with the 3.0 code train. ASICs limitations are specific to a policy direction rather than being a global number. The items above will return how many rules have been created and how many are available in each direction.

```
get pol disable
```

This will display only the policies that have been disabled.

## rms

```
get rms <arguments>
<return>    list rms information
ctx         list all rms contexts
```

View RMS internal information, including context limits.

## session

```
get session info
```

Display only the summary header of the 'get session' command. It is helpful for scripting where output only lists current, maximum, and failed sessions.

## snoop

```
snoop <arguments>
<return>      turn on snoop
direction     snoop direction
ethernet      snoop specified ethernet
info          show snoop information
interface     snoop which interface
ip            snoop ip packet
off           turn off snoop
```

Snoop allows you to sniff traffic on any firewall interface. Take caution when using this, and use in conjunction with the 'set console dbuf' and 'get dbuf' commands if possible!

## sys-cfg

```
get sys-cf
```

Display almost every system internal limit imaginable. This is quite helpful to determine the maximum number of entries allowed in any give system parameter. Executing this on different platforms will return the system limits appropriate to that hardware and software platform.

## system

```
get system scale
```

View basic system limits including maximum entry size and maximum count on: ASICs, Addresses, Sessions, Routes, Users, IPSEC VPNs, Mapped IPs, and policies.

## tcp

```
get tcp
```

Display information regarding system sockets. This is a tad more detailed than 'get socket' but probably not as concise or helpful. Extremely detailed information can be obtained from each individual socket by specifying a socket id number with either command. This is not listed in deprecated status because the output of 'get socket' is slightly different and includes udp information as well.

## undebug

```
undebug <arguments>
```

This command will disable debug output for the specific argument.

```
undebug all
```

Quickly turn off all debugging; don't leave debugging on indefinitely because it slows the box way down.

## vpnmonitor

```
set vpnmonitor frequency <time>
```

Modify the VPN monitor frequency timer to improve VPN failure detection times

## *To Be Determined*

The following commands are pending further research and dissection. However, they have been placed here for your review and enjoyment. If you have any comments on their function or potential use, please feel free to send your comments and join the folks in the credits section!

```
set tail-route
get arp count
get net-buf
get pport count
get pport dst <ip>
get break
set break <args>
get chunk table
set|get dummy
get dump <args>
get icmp
get ip-frag
get module
set flow no-frag
get nat <cookie|registry>
get pool
get rtd detail
get summary <reset|src>
get tty
set|get traffic gbl
set|get traffic mbl
get traffic history
get traffic interval
get int null
set int null ping
set int self <args>
get int self
set int <int> id
set mem <bytes>
set|get portnum tdp|udp
set priv <num>
set psc <num>

ns25-> set ppc ?
int set PPC INTERNAL egisters
mem set memory
pci set PPC PCI egisters
```

## Conclusion

The purpose of this document was to present a fairly exhaustive list of undocumented ScreenOS commands. This document was written to increase the Netscreen firewall administrator's system

knowledge and his or her ability to troubleshoot issues on the CLI. It has focused primarily on commands that are **new**, **custom made**, and part of the **engineering** toolkit to provide a documented reference for system options and troubleshooting parameters that would otherwise be difficult to uncover. Security administrators should fully understand the risks involved in attempting to make use of the contents of this document before tackling issues in a production environment.

## References

[1] Netscreen Mailing List Archives  
<http://www.gorbit.net/nn/index.html>