Object Filler & Object Dumper
Version 2.0

User's Manual

Martín Humberto Hoz Salvador
mhoz@mexico.com

February 12th, 2005
Revision B

TABLE OF CONTENTS

Disclaimer, License of use and Limit of Liability

The programs Object Filler, Object Dumper and its companions described on this documentation are not official nor supported in any form, expressed or implicit, by any entity.

The above statement includes that Check Point Software Technologies does not support nor backs these tools. These tools were made by a merely personal initiative as a technology proof-of-concept, neither for profit nor for financial gain in any form.

These programs come with no support of any kind and with no warranty of any kind. You (the user of it) are responsible for any use these programs may have, good or bad; and for any other good or bad thing derived of this program anyhow while you use it directly or indirectly (including but not limited to data loss, programs failure of any kind or promotions). The author(s), their employers and any other entity with direct or indirect relationship to them, are not liable in any way or form because of the misuse, abuse or use of the mentioned programs. The author(s) have made no warranty nor representation that the operation of this software will be error-free or suitable for any direct or indirect application, and they are under no obligation to provide any services, by way of maintenance, update, or otherwise. This software is an always experimental prototype offered on an as-is basis; and should be treated accordingly.

You are not allowed to disassemble, reverse engineer or perform any other known or unknown way to discover the mechanisms used by the tools, over the binary forms. You may also not use these tools, the knowledge of them, to harm in any known or unknown way to the author, Check Point Software Technologies, or any other entity with relationship to them.

You may use the programs Object Filler, Object Dumper and its companions described on this documentation, free of charge (free of cost), but if you are not a Check Point Software Technologies employee, you cannot redistribute them in any form. Check Point Software Technologies employees are allowed to redistribute this software, as long the original program package and documentation is preserved. If you want to share these programs, you must direct the interested entity to download it from any of the sites where they are available.

All other brands and commercial trademarks mentioned on this documentation, and in the programs described by this documentation, are property of their respective owners.

Using the tools in any way, makes you implicitly accept the terms of use above listed.

Introduction

Administrators using Check Point products and maintaining the SmartCenter server, may find the graphical interface provided to manage their security configuration very appealing for the day-to-day operations. However, from time to time there are needs where a graphical interface may fall a bit short, or where you may wish additional functionality today not freely available. Samples of these cases may be:

1.- You are tasked by a customer to configure a Check Point solution where creating all the networks from 10.100.0.0 to 10.100.255.0 with a 24 bit netmask (255.255.255.0) is needed. This means you'll have to manually click to create 256 objects. If all of them will be NATted using Hide NAT and the IP address 1.2.3.4, them more clicks are needed when generating the objects. This may be painful and time consuming

2.- You're migrating from a Cisco PIX or Cisco Router ACL, Juniper/NetScreen, Symantec's Raptor, SecureComputing's SideWinder or Gauntlet to a Check Point solution. You have tons of rules already created and you want to save some time while passing this information from the old platform to your brand new Check Point.

3.- You or your customer has 50 IP addresses (or more) assigned to internal users or services, on a Microsoft Excel sheet. Those IP addresses are not on any given order, so you cannot create networks or ranges, and it's needed to create individual host objects representing such workstations for granular access control policies. This may take some time and can be error-prone if done via a Graphical Interface.

For all those cases, it's possible to save some time using a marvelous command line interface that is installed with all the Check Point SmartCenters on any supported platform. This command line interface is supported by Check Point and is called dbedit.

dbedit is then, a very powerful command-line based tool that allows to control the Check Point's SmartCenter object database (which you can find in a somehow human-readable text file on $FWDIR/conf/objects_5_0.C on SUN Solaris, Nokia IPSO, Red Hat Linux, Check Point SecurePlatform, or %FWDIR%\conf\objects_5_0.C in Microsoft Windows)

Object Filler and Object Dumper are tools oriented to Security Administrators or Security Engineers using Check Point Products, which ease the use of dbedit to perform administrative tasks

Object Filler is an automated tool that can take a couple of IP addresses and a netmask as entry from the command line, or information from a file with certain format, and then produces dbedit commands that automatically generate network objects, services and rules, easing the task of populating the SmartCenter with the information you need.

In the other hand, Object Dumper does somehow the opposite: given an Objects_5_0.C file, Object Dumper can export the content to a CSV file which you can review or modify using any spreadsheet program able to understand CSV (Comma Separated Values). Microsoft Excel is an example. If you want, you can modify such CSV file, and then import back your modifications to the SmartCenter Server using Object Filler.

Object Dumper also gives you the chance of exporting the Objects_5_0.C file to an HTML table for documentation purposes, even though I would recommend using some other program to do that (please refer to the FAQ section of this document for a list of suggested programs).

This manual describes how these tools work. Please read this documentation before sending a question to the author or any other entity, such as mailing lists.

Contacting the Author

If you wish to contact the author of the tools, you may send an e-mail to mhoz@mexico.com with the subject *About Object Filler*. I always read and answer my e-mails, and I always take in account the suggestions given.

If for any reason, the communication requires to transfer some sensitive files or messages, you may encrypt such content using my PGP public key included in the distribution files. If for any reason you have no access to such file, you may locate it in a public key server, under the PGP Key ID 0x0454E8D9

I strongly encourage you to send me an e-mail if you are successfully using these tools in your environment. So far, the amount of feedback I get is very low, and would like to know more of any use you may have given to the tools, any stories (good or bad) around them, suggestions of course, and overall what do you think on them in general.

Related programs

There are several programs that do tasks similar or related to what Object Filler and Object Dumper do. Some of them are supported by Check Point. The following is an incomplete list of such programs, that you may find useful in tasks Object Filler/Object Dumper are not designed to do:

- **Web Visualization Tool:**
  http://www.checkpoint.com/techsupport/downloadsng/utilities.html#visualization
  Officially supported by Check Point, that supports exporting rules and objects to HTML/XML format.
- **FW1Rules:**
  http://www.wyae.de/software/fw1rules/
  Unsupported by Check Point tool, that allows to export both objects and rules in several formats, including HTML and CSV. Written on PERL.
- **CPrules:**
  http://www.wormnet.nl/cprules/
  Unsupported by Check Point tool that allows to export the rulebase and objects to HTML. Tool especially enhanced for NG installations. Written on PERL.
- **Upgrade tools:**
  http://www.checkpoint.com/techsupport/downloadsng/utilities.html#upgrade_verify
  Supported by Check Point tools that help to migrate from previous versions to the current one. These tools also proactively tell of potential problems (and the solution) when upgrading. Besides that, they can export and import the whole configuration in different machines and even across platforms. Documentation included in the package.
- **CP Merge:**
  http://www.checkpoint.com/techsupport/downloadsng/utilities.html#cpmerge
  Supported by Check Point tools that help import, export and merge policies from and to the SmartCenter. Documentation included in the package.
- **fw dbimport/fw dbexport:**
  Supported by Check Point command line tools included in the product, that help to import and export user's information. The documentation for such commands is available at the Docs directory in the CD where the Check Point's software comes, and the filename is CommandLineInterface.pdf. You may find the Online version of the manual here (Valid Software Subcription Account is required):
  http://www.checkpoint.com/support/downloads/docs/firewall1/r55/CommandLineInterface.PDF
  http://www.checkpoint.com/support/downloads/docs/firewall1/r54/CommandLineInterface.pdf

Object Filler & Object Dumper Programs Availability

Object Filler and Object Dumper have no associated cost (i.e. no license fee) – as they are not part of the official Check Point Software.

Latest versions are always available at the following sites (they are not listed on any particular order):
http://www.phoneboy.com - under *FireWall-1 FAQs* and then *Software Downloads*
http://lindos.dnsalias.com/
http://www.lindercentral.com/ofiller/
http://mhoz1.sofaware.net/ofiller/ - Unreliable Server

These programs are also available through Check Point System Engineers world wide on a non-official way - i.e. they are not obligated to provide it to you, they are also not obligated to support you if you run into problems. They are not even obligated to know that these tools exist.

I would like to thank very, very, very (very) much to Jeff Mousseau, Nokia SE at Canada, for letting me use his website (http://www.digitalmigrations.com) as the official download site for the tools for two years (2003-2004). Such a reliable and well organized website for sure will be missed by the security community worldwide.

I also want give my sincere and deep thanks to Brian Linder, Pedro Paixão and Dameon D. Welch-Abernathy (a.k.a. phoneboy) for hosting the latest version of the tools in their websites.

Acknowledgements

I would like to thank the following individuals for their direct and indirect help with this: providing information to me, testing it, telling me it actually works! (or that actually have bugs) ;-), commenting and suggesting features or utilization scenarios, and overall helping me on improving both programs – I'm sure I'm still missing people, so I apologize in advance for that.

Adrián Espinosa, Andrew Singer, Amir Kossover, Arturo Gómez, Brian Linder, Chris Lyttle, Chris Tobkin, Dameon D. Welch-Abernathy, Damien DeVille, David Hernández, Diego Lastra, Dino Constantinou, Elad Lavi, Eli Faskha, Emilio Sánchez, Enaela García, Erez Shtang, Fernando Acosta, Gil Sudai, Gil Shapira, Héctor Garza, Idan Plotnik, Jaime Castañeda, Jeff Mousseau, Jim Hebert, José Agüero, Juan Garza, Julio Salas, Kellman Meghu, Kris Boulez, Leonid Belkind, Marc Gorelick, Marc Lampo, Mario Cinco, Mario Garibay, Mats Ekdahl, Ofer Barzvi, Ofir Barzilai, Oscar Viniegra, Paul Frumer, Pedro Paixão, Peter Sandkuijl, René Tavera, Rickardo González, Rob Sparre, Rodrigo Díaz, Ronen Leshem, Sharon Besser, Tal Shevach, Tom Calarco, Udo Schneider, Zohar Erel.

To all of you, my deepest and sincere thanks for helping me to bring the tools to the current stage in one way or another.

Program History

* Version 2.0 – February 2005

- Object Filler
  - Added support for ICMP, Other, RPC and DCE-RPC services in CSV files.
  - Added support for Check Point Dynamic Gateways (Check Point Gateways with Dynamic IP address) in CSV Files
  - Enhanced support for Raptor files, including services recognition
  - Added support for interfaces on Check Point gateways (including Clusters and Dynamic IP gateways) and interoperable devices. Interfaces with dynamic IP are also supported. This is in CSV files.
  - Added support for setting the encryption domain on Check Point gateways (including clusters and dynamic IP gateways) and interoperable devices in CSV files.
  - Added support for setting the WebServer property, when configuring Hosts in CSV files.
  - Enhanced and corrected bugs in the support for rules from CSV files
  - Added support for a "modifications mode", used mainly to modify some properties on currently build objects in CSV files. This option is designed to be used with object dumper.
  - Enhanced support for NATted objects when importing from PIX configurations. Now the results are more accurate.
  - Enhanced support for Rules processing when importing from PIX and Cisco Router configurations: Now splits in different sections of the imported rulebase, the different ACLs that the configuration may have
  - Now, when creating groups from CSV Files the groups are created first and then elements are added to them.
- Object Dumper
  - Added support for services (TCP, UDP, ICMP, Other and RPC)
  - Added support to recognize basic rules from the rulebases (rulebases_5_0.fws) file.
  - Added support for interfaces with Check Point gateways (including clusters and dynamic IP gateways) and Interoperable devices. This includes interfaces with dynamic IP
  - Support for recognizing the webserver property on hosts
  - Support for recognizing the encryption domain on Check Point Gateways (including clusters and dynamic IP gateways) and Interoperable devices.
  - Enhanced the support for reading objects.C files from gateways.
  - Tabulation (-tab) mode is not supported anymore. (seems that nobody was actually using it)
- Documentation
  - For the first time, the tools have a more decent manual. It's the plan to enhance the documentation in the next releases

* Version 1.9.2 - November 2004

- Object Filler
  - Added support for TCP/UDP services when importing from CSV Files, Cisco PIX, Cisco Routers and NetScreen configuration files.
  - Added support for importing basic layer 3 and layer 4 rulebases from CSV files
  - Added support for importing rules from Cisco PIX and Cisco Routers with access-list configurations.
- Object Dumper:
  - Added support for objects.C files found on gateways, for recovery options.

- Added support for Check Point dynamic IP gateways.

* Version 1.8 - March 2004

- Object Filler
  - Support to DELETE and RENAME operations over objects, when specifying such operations on CSV files
  - Support to Domain and Dynamic objects.
  - Clusters and cluster members are also now supported.
  - Support to the following SmartLSM objects: IP40 ROBO gateways, Edge X ROBO gateways, profiles (SmartLSM VPN-1 Egde/Embedded NG profiles).
  - Support to PIX network-object groups and to name statements.
  - Support to Raptor configuration files.
  - Support to Cisco IOS ACLs (including the ones declared with inverse masks).
  - Support to NONAT mode when importing from files.
- Object Dumper
  - Added support to clusters and cluster members.
  - Added support to Domain objects.
  - Added support to SmartLSM VPN-1 Edge/Embedded profiles
  - Added support to Dynamic Objects.

* Version 1.6 - January 2004

- Object Filler
  - Fixed bug: Now it works with IP address range objects behind a NAT IP range.
  - Fixed bug: Now it recognizes "cpgws" as a valid object type when using command line parameters.
  - Fixed bug: Correctly handles objects hidden behind "All gateways", when "All" is specified on the NATting Object column, when importing from CSV files.
  - Enhanced interactive mode with more comments to ease user experience.
  - Improved support to NAT statements when importing from PIX.
  - Improved summary information when importing information from any file.
  - Now groups are supported when using the cgi mode (HTML form) and importing from CSV files.
  - Support to import from SideWinder configuration files.
  - New output mode (ASCII) introduced, which instead of writing dbedit commands, leaves the information on CSV format, which is easier to read and compare.
  - Support to Interoperable Devices, Plain Gateways and OSE Devices when importing from CSV files and when generating objects from command line.
  - Now when importing from a LIST type of file, it's possible to create ranges and groups.
  - Native binary support for Linux/SecurePlatform.
- Object Dumper
  - Now it recognizes Interoperable Devices, Plain Gateways and OSE Devices.
  - Enhanced interactive mode with more comments to ease user experience.
  - HTML mode for output files is now available.
  - TAB (Column) mode for output files is now available.
  - Native binary support for Linux/SecurePlatform.

* Version 1.4 - December 2003

- Support to interactive mode (command line is still supported).
- Enhanced support to duplicates (now it takes in account the object type, not only the IP address).
- IP ranges support when creating objects and when importing from CSV files.
- Support to import objects from Gauntlet configuration files.
- Support to comments when importing from CSV files and List files.

- Support to import groups when importing from CSV files.
- Support to groups when importing from CSV files.
- Gives a summary of how many objects of each known type were processed.
- **Object Dumper companion tool was created**.

* Version 1.2 - July 2003
  - Support to import objects from CSV (comma separated) files, where you can detail the objects you need to create.
  - Support to import from lists files, where you just detail IP and netmask,
  - Object name and everything else is calculated automatically.
  - Support to import objects from operating system's host file.
  - Support to import objects from Cisco PIX and NetScreen configuration files, and create network objects from there. Importing rules is not supported.
  - Support for Hide NAT on created sequential objects.
  - Support for Static NAT on imported objects from files.
  - Support to NAT ranges to hide created objects.
  - Support for color specification on new objects.

* Version 0.96 - May 2003
  - Support for importing from CSV files.
  - Support to Check Point Host and Check Point Gateway objects.
  - Support to Hide NAT for objects generated using command line.
  - Support for colors on created objects.

* Version 0.8 - April 2003
  - This is the Initial Public release ("First Customer Shipment").
  - Supported only creating hosts and networks.

* Version 0.5 - April 20th, 2003 – 06:57 AM
  - This idea of Object Filler is born. I started with some preliminary designs, algorithms and coding that morning.

Known limitations, issues and particular behavior for both programs

Object Filler's known limitations and issues

- Object Filler cannot detect if there's enough space on disk for the output file. So, please be careful and send program's output to a file placed on a disk or partition with enough space for this output. For each object created, take an average size of 750 bytes each. Usually it will take less space, but using this consideration you will be safe.
- When importing from Cisco PIX configurations, IP Address ranges specified on the original configuration won't be processed. If you've a range like 1.2.3.4-1.2.3.99, this range will simply be ignored. Individual IP's will be processed, however.
- When importing from Cisco PIX configurations, and there are 2 global statements (one for the external interface and one for the DMZ, as an example), only the first one found is applied and the second is ignored.
- When importing from Cisco PIX configurations, the use of names in access-list statements is not supported. The import of an access-list that contains names will result in a problematic rule.
- When importing from Cisco PIX configurations, the import of service groups is not supported.
the use of names in access-list statements is not supported. The import of an access-list that contains names will result in a problematic rule.
- When importing from NetScreen configurations, Hide NAT and PAT are not supported, so if you have "dip" or "vip" statements, they will be ignored as NAT statements but will be processed as any other normal line (i.e. the IP addresses will be converted to objects) . Please note that if you have "mip" statements (Static NAT), they will be processed as Static NATted objects, and the right output will be produced.
- When importing from NetScreen configurations, the import of service groups is not supported.
- When importing from SideWinder configurations, domains and service groups are not supported while working over the ACL tables.
- 0.0.0.0 is not a valid IP address for Object Filler.
- When importing from files, due file internal representation, sometimes the last line of the file will be processed twice. However, the output for this line will be produced just right (once), since Object Filler detects and avoids duplicates by default.
- When importing groups from CSV files, it's not possible to specify color nor comments for them. This is planned to be fixed, but currently not supported
- It's not possible to include groups as members of other groups, when importing LIST files with Object Filler.
- OSE devices cannot have NAT properties defined.
- It's not possible to use Check Point Hosts, Check Point Gateways, Check Point Dynamic Gateways, OSE Devices, Interoperable Devices, Plain Gateways or Services, when importing LIST files with Object Filler.
- When defining interfaces, the interfaces have to be defined after the gateways that owns such interfaces. Also, Object Filler assumes that it's defining the first interfaces on the object. Adding interfaces to an object that already has interfaces defined is not supported.
- When creating rules from CSV Files, adding rules to an existing policy is not supported.
- When processing rules from CSV Files, using User Groups as sources (user@location) is not supported
- When processing rules from CSV Files, using resources for services (uri resources, smtp resources, etc.) is not supported

- When Object Filler is processing rules, only Accept, Drop and Reject are accepted as valid actions. Other actions are not supported
- The tool has been not tested with VSX management.

Object Dumper's known limitations and issues

- Only the following object types are supported: Check Point Hosts, Check Point gateways, Check Point Dynamic Gateways, Check Point Clusters, Check Point Cluster Members, plain hosts, networks, IP Address ranges, OSE Devices, Plain Gateways, OSE Devices, Domains, Dynnamic Objects, Interfaces and Groups. TCP, UDP, ICMP, RPC, DCE-RPC and type Other Services are supported. Other network object types and services are currently not supported.
- Object Dumper does not support SmartLSM VPN-1 Edge/Embedded ROBO gateways, like IP40 or Edge X SmartLSM ROBO gateways.
- Comments and colors for groups are not fully supported.
- Currently, if you want to process a rulebases file, you must process an objects file also. Processing just the rulebases file by itself is not currently supported.
- When processing rules from rulebases_5_0.fws, using User Groups as sources (user@location) is not supported. Instead, the name of the network (the location part of the source) is showed.
- When processing rules from rulebases_5_0.fws, the resources for services (uri resources, smtp resources, etc.) are not recognized properly.
- When processing rules from rulebases_5_0.fws, only Accept, Drop and Reject are accepted as valid actions. Other actions are not supported.
- When processing rulebases, if the rulebases_5_0.fws contains several policy packages (i.e. policies with different names), these are exported as rule section headers (not different rulebases), and recognized by Object Filler as sections, not as different policies.
- When exporting objects with Object Dumper, due some limitations on the program, some times the color may be exported wrongly. On these cases, the object's color will be reset to black.
- Object Dumper was not tested on this version with Provider-1.
- The tool has been not tested with VSX management.

Tested scenarios

These programs are not official by Check Point Software Technologies, nor supported in any way by any entity.

These programs can run in a native form and have been tested by the author and others on:

- Microsoft Windows NT 4.0 and Microsoft Windows 2000.
- Red Hat Linux 7.2
- SecurePlatform Next Generation with Application Intelligence R55 (NG+AI R55)

However, since program's output is just regular ASCII TEXT, it can be easily transferred (using scp, ftp, diskettes or any other file transfer mechanism) to a SmartCenter Server on other operating system, as shown below. Please **make sure** that if you transfer files among operating systems you do that as ASCII TEXT files.

Since the tools are programmed under windows, they are tested more deeply in this platform. Then they are recompiled under GNU/Linux and tested for basic functionality. While this means not so much testing on GNU/Linux, the functionality should be the same. I recently got report of somethings working in Windows but not in Linux. If it happens that you find something like this, I'd appreciate very much your report on it.

Requests for natively supporting other OSes by the programs are always welcome and taken in account. Solaris SPARC is in the roadmap for future releases, but not any firm dates yet.

Object Filler's output has been tested so far with SmartCenter Servers running on:

- SecurePlatform: NG+AI R55
- Nokia IPSO: NG+AI R55
- Windows NT 4.0: NG FP3
- Windows 2000 Server: NG+AI R54, NG+AI R55

Object Filler's output has also been tested with Provider-1 NG+AI R54, MDS Manager and Container Server, under Solaris 2.8

To review what specific software/firmware versions were tested when importing configurations from other brands, please take a look on the proper section below on this document.

Object Dumper has been tested using Objects_5_0.C files coming from:

- SecurePlatform: NG+AI R54, NG+AI R55
- Nokia IPSO: NG FP3, NG+AI R54
- Windows: NG FP3, NG+AI R54, NG+AI R55

As you may imagine, my testing resources are finite and small, so if you use this program on a different environment, I'd really appreciate if you send me a note saying so, will surely help improving this documentation, and everybody that uses these programs.

In the event of any program bug, failure, request, comment, grammar correction on this documentation or the messages sent by the programs, or any other request for enhancement, **PLEASE** send an e-mail to ***mhoz@mexico.com*** with the subject *About Object Filler*. I don't promise I'll fix immediately whatever you're asking for, as I'm doing this on my -not so abundant- free time, but I'd like to hear from you anyways, and I promise to try to implement your suggestion, whatever that is. In general, so far I've been able to implement specific features, in a two-week timeframe, for people asking/requesting me such features and willing to help on testing them.

If you wish, you may use the provided public PGP key (included in the distribution file) to encrypt any files or message directed to the author.

Always keep in mind also that this is a not official nor supported software.

Tools Installation

The tools do not need to be installed at all. They are executables that don't need any special library. Under Un*x flavors, the tools need access only to the standard libraries that any other program needs.

In the other hand, remember that you don't have to have the executable running natively on the platform where you have the SmartCenter server. If you have the SmartCenter Server on Nokia IPSO for example, you could execute Object Filler or Object Dumper in any other platform, by just transferring the needed files via FTP, SCP or any other file transfer mechanism.

Introduction to the object types used by the tools

Introduction

All the elements from which the a network security policy are represented by *Objects* in SmartCenter. Each object is an atomic element that has different properties. You may see an introduction of this in the SmartCenter manual that comes on the CD where the Check Point software is distributed. The Chapter 1 (SmartCenter Overview) contains a section named "Managing Objects in SmartDashboard", as well as the Appendix A named "Network Objects" where you may find more information with regards to Objects.

The following is a brief and simple explanation on common situations for the objects, and what to expect or when to use them while using Object Dumper and/or Object Filler.

Network Objects Definition

Check Point Hosts
Represent servers with one or more NICs (interfaces) attached to them, but where no routing through it is performed (packets cannot go from one interface to another). If a Check Point Host has more than one interface, all of them will be automatically marked as external. Check Point Hosts usually indicate VPN-1/FireWall-1 SecureServers or SmartCenter Servers that are in distributed configuration.

Check Point Gateways
Represent gateways - i.e. hosts with more than one interface, but where the packets are processed, routed and passed (if allowed) between interfaces. Usually indicate Check Point Gateways (Enforcement Points), either StandAlone or distributed configuration.

Check Point Dynamic Gateways
These are Check Point Gateways, where the main IP address is a dynamic IP Address (i.e. is not a Fixed IP Address)

Check Point Cluster
A group of Check Point Gateways that behave as if they were just one entity.

Check Point Cluster Member
A Check Point Gateway that belongs to a cluster.

Plain Host
A host with one or more interfaces, where no Check Point product is installed, and where no routing among interfaces is performed. Used to represent user's machines, workstations, hosts or servers.

Network
A simple network segment, delimited by an IP address and a netmask.

Plain Gateway
A device that passed packets through it – i.e. performs routing among interfaces, but has no Check Point products installed.

Interoperable Device
A Plain Gateway that has some sort of VPN software installed on it, and has the capability of doing VPN with a Check Point Gateway.

OSE Device
Is a Cisco, Nortel or 3Com device from which is possible to read and/or write security rules.

IP Address range
A group of IP addresses that cannot be delimited with a netmask, and that for the effects of a security policy, behave like one.

Dynamic Object
An object that takes different IPs, depending of the associations made at the gateway level, once the security policy has been applied.

Domain
A domain name.

Network Object Group (Simple Group)
A set of Network objects.

Interface
An interface that belongs to a gateway.

SmartLSM Profile for NG Embedded
A Profile going to be used to define security policies and VPNs with NG Embedded devices, with Smart Large Scale Manager (LSM). Valid only if SmartLSM has been enabled on the SmartCenter.

SmartLSM Edge X gateway
A gateway defined as Check Point VPN-1 Edge X gateway, usable with SmartLSM. Valid only if SmartLSM has been enabled on the SmartCenter.

SmartLSM IP40 gateway
A gateway defined as Nokia IP40 gateway, usable with SmartLSM. Valid only if SmartLSM has been enabled on the SmartCenter.

Services Definition

TCP Service
Service definition that uses the Transmission Control Protocol (TCP). The main property for this is a Port number that may go from 1 to 65,535

UDP Service
Service definition that uses the User Datagram Control Protocol (UCP). The main property for this is a Port number that may go from 1 to 65,535

ICMP Service
Service definition that uses the Internet Control Message Protocol (ICMP). The main property is the ICMP type, as defined in several RFCs.

RPC Service
It is a definition that makes reference to service running over Remote Procedure Calls. The definition of the service is made with program numbers.

DCE-RPC Service
Distributed Computing Environment/Remote Procedure Call. It's a different kind of RCP services. The identification is made using UUIDs.

Table of Objects, Services and Operations supported by Object Filler

	CLI mode	CSV File create	CSV File modify	Supports NAT	Webserver property	Encryption Domain
Network Objects						
Check Point Host	ss	ss	modss	Yes	No	No
Check Point Gateway	cpgw	cpgw	modcpgw	Yes	No	Yes
Check Point Dynamic Gateway	N/S	dynamicgw	moddynamicgw	No	No	Yes
Check Point Cluster	N/S	cluster	modcluster	No	No	Yes
Check Point Cluster Member	N/S	member	N/S	No	No	No
Plain Host	host	host	modhost	Yes	Yes	No
Network	net	net	modnet	Yes	No	No
Plain Gateway	plaingw	plaingw	modplaingw	Yes	No	Yes
Interoperable device	idevice	idevice	modidevice	Yes	No	Yes
OSE Device	ose	ose	modose	No	No	No
IP Address Range	range	range	modrange	Yes	No	No
Dynamic object	N/S	dynamic	N/S	No	No	No
Domain	N/S	domain	N/S	No	No	No
Network Objects Group (Simple)	N/S	group	N/S	No	No	No
Interface	N/S	interface	N/S	No	No	No
SmartLSM Profile for Embedded NG	N/S	lprofile	N/S	No	No	No
SmartLSM VPN-1 Edge X gateway	ledge	ledge	N/S	No	No	No
SmartLSM IP40 gateway	lip40	lip40	N/S	No	No	No
Services						
TCP Service	N/S	tcp	modtcp	N/A	N/A	N/A
UDP Service	N/S	udp	modudp	N/A	N/A	N/A
ICMP Service	N/S	icmp	N/S	N/A	N/A	N/A
RPC Service	N/S	rpc	N/S	N/A	N/A	N/A
DCE-RPC Service	N/S	dcerpc	N/S	N/A	N/A	N/A
Other Service	N/S	other	N/S	N/A	N/A	N/A
Services Group	N/S	srvgroup	N/S	N/A	N/A	N/A
Operations						
Delete operation	N/S	DELETE	N/A	N/A	N/A	N/A
Rename operation	N/S	RENAME	N/A	N/A	N/A	N/A

N/S=Not Supported. N/A=Doesn't apply
* CLI mode means it is supported by Object Filler from Command Line. The content of the cell is the keword used.
* CSV File create means that object can be created via a CSV File.
* CSV File modify means that the object exist, but its properties will be modified.
* WebServer property means that the object will be marked as webserver
* Encryption domain means that the object will have an encryption domain associated.
* Operations Delete and Rename are only supported for Network Objects, not services.

Table of Objects, Services and Operations supported by Object Dumper

	Supported	Interfaces	Encryption Domain
Network Objects			
Check Point Host	ss	No	No
Check Point Gateway	cpgw	Yes	Yes
Check Point Dynamic Gateway	dynamicgw	Yes	Yes
Check Point Cluster	cluster	No	Yes
Check Point Cluster Member	member	No	No
Plain Host	host	No	No
Network	net	No	No
Plain Gateway	plaingw	Yes	Yes
Interoperable device	idevice	Yes	Yes
OSE Device	ose	No	No
IP Address Range	range	No	No
Dynamic object	dynamic	No	No
Domain	domain	No	No
Network Objects Group (Simple)	group	No	No
Interface	interface	No	No
SmartLSM Profile for Embedded NG	lprofile	No	No
SmartLSM VPN-1 Edge X gateway	N/S	No	No
SmartLSM IP40 gateway	N/S	No	No
Services			
TCP Service	tcp	N/A	N/A
UDP Service	udp	N/A	N/A
ICMP Service	icmp	N/A	N/A
RPC Service	rpc	N/A	N/A
DCE-RPC Service	dcerpc	N/A	N/A
Other Service	other	N/A	N/A
Services Group	srvgroup	N/A	N/A

N/S=Not Supported. N/A=Doesn't apply
* Supported means that Object Dumper will recognize the object, and the output will have the keyword listed.
* Interfaces means that if the object has interfaces attached, such interfaces will be listed in the output.
* Encryption domain means that if the object has an encryption domain associated, it will be listed in the output.

Object Filler

Program syntax

```
1) ofiller  help  (prints help pages - with examples)
2) ofiller  -f file -i input [-o|-a] file [-c color] [-t type]
            [-p policy] [-nopv] [-nonat] [-v]
3) ofiller  -t type -s ip -d ip -m mask [-c color]
            [-n ip | -ns ip -nd ip -nm mask] [-b obj] [-o|-a] file [-v]
```

Program syntax #1 : Asking for help

```
ofiller help
```

This syntax allows you to see the incorporated help in the program. The result is simply a brief documentation on the program's switches

Program syntax #2 : Importing from a file

```
ofiller  -f file -i input [-o|-a] file [-c color] [-t type]
         [-p policy] [-nopv] [-nonat] [-v]
```

This syntax allows you to produce objects information (and possibly rules information) having as source for it, a file. This file may be a CSV File in a pre-defined format, or the configuration of another firewall.

-i - Input type - It can be either:

* csv - File must be formatted on csv format. You may take a look on the file sample_csv.csv for more information on this switch.

* list - File must contain 2 mandatory fields: IP Address and netmask. You may take a look on the file sample_list1.csv for more information.

* hosts - File has the format of a hosts file (/etc/hosts on Un*x systems, or %SYSTEMROOT%\\system32\\drivers\\etc\\hosts on Microsoft Windows systems). You may take a look on the sample file sample_hosts for more information.

* pix - File is the configuration listing from a Cisco PIX device. You can get this information from a PIX device using the command \"show running\" or \"write terminal\".

* netscreen - File is the configuration listing from a NetScreen device. You can get this information from a NetScreen device using the command \"get config all\".

* gauntlet - File is the configuration file of Gauntlet (gauntlet.conf).

* sidewinder - File is the configuration file of SideWinder (ACL and ipfilter files).

* raptor - File is the Raptor configuration file that contains the IPs and rules of the firewall.

* ciscorouter - File contains is the result of executing the \"show running\" command from a device running Cisco's IOS.

This is a required parameter.

Example:

```
ofiller -i csv -o all_objects.txt -f input.txt
```

-f - input File - Takes the input from the specified file. See details on –i switch on how this file needs to be formatted. **Required parameter.**

Example:
```
ofiller -f my_old_5XT.cfg -o output.dbedit -i netscreen
```

-p - Policy name - It specifies the policy name that the imported policy will have when it's imported into the SmartCenter. It also used to tell Object Filler that you want to import a policy. If you don't specify this switch, even if the configuration contains a policy, Object Filler won't try to process it. This switch is only valid for the supported input files, currently Cisco PIX, Cisco Routers and CSV Files. **Optional parameter**.

Example:
```
ofiller -p mypolicy -f Conf_PIX515.txt -o output.dbedit -i pix
```

-nopv - No Policy Verification - This switch is used to decide if the objects will be verified to see if they were processed by Object Filler before or not. This switch is relevant only if used with -p switch. If it is not specified, all the objects present in rules definition that have not been processed, will be translated to \"Any\". Use this switch if you are processing only rules, but no objects. **Optional Parameter.**

Example:
```
ofiller -nopv -p mypolicy -f new_rules.csv -o output.dbedit -i csv
```

-c - Color - The color we'll use to build the objects. Can be black blue, green, gray, red, pink, brown, cyan, yellow, orange, magenta, sienna, gold, coral, firebrick. When importing from a File, this parameter will take precedence over any specified color on the importing file. **Optional parameter.**

Example:
```
ofiller -c blue -f c:\files\my_old_535.txt -i pix -o d:\\tmp\\objects.txt
```

-t - object Type - It's the object type we'll build can be host, cpgw (Check Point Gateway), ss (SecureServers - Check Point Host), idevice (Interoperable Device), plaingw (Plain Gateway) or ose (OSE Device). This parameter will be relevant only if importing from a hosts file. If specified with any other type of file, it will be ignored**. Optional parameter**.

Example:
ofiller -t ss -f /etc/hosts -i hosts -o /home/admins/root/host_smc.txt

-o - Output file - The name of the file where resulting dbedit commands will be stored. Please make sure you have enough disk space to store all produced commands. To calculate this pace, take an average of 750 bytes per object to process. The File must not exist previously.. If it exists, the execution of the program will be aborted. This switch is mutually exclusive with -a (i.e. if you can only specify -a or -o, but must at least use one of them). **Required parameter if –a was not specified.**

Example:
```
ofiller -o all_objects.dbedit -i csv -f input.txt
```

-a - Ascii file - The name of the file where resulting CSV information will be stored. This is an alternative to -a, and instead of writing dbedit commands, Object Filler writes information regarding the created objects (name, ip, comments, etc.) on CSV format, so you can take a look on a spreadsheet program first. Please make sure you have enough disk space to store all produced commands. To calculate this space, take an average of 120 bytes per object to process. File must not exist previously. This switch is mutually exclusive with -o (i.e. if you can

only specify -a or -o, bust must at least use one of them). **Required parameter if –o was not specified.**

Example:
```
ofiller -a all_objects.csv -i netscreen -f ns5200.conf
```

-nonat - Use this option if you want that the importing file NAT statements not to be processed. This will cause that build objects won't be NATTEd, even if they are on the original configuration used to feed Object Filler. **Optional parameter.**

Example:
```
ofiller -a all_objects.csv -i netscreen -f ns5200.conf -nonat
```

-v - Verbose mode - shows on the console (the screen) details on how the processing is being done line-by-line. This is very useful especially when importing files, since it says how each line was treated. **Optional parameter.**

Example:
```
ofiller -v -i csv -o all_objects.txt -f input.txt
```

Program syntax #3 : Specifying arguments from command line

```
ofiller  -t type -s ip -d ip -m mask [-c color]
         [-n ip | -ns ip -nd ip -nm mask] [-b obj] [-o|-a] file [-v]
```

-t - object Type - It's the object type we'll build. It can be:

- cpgw (Check Point Gateways)
- ss (Check Point Hosts)
- host (Plain hosts)
- plaingw (Plain Gateways)
- net (Plain Networks)
- range (IP Address ranges)
- ose (OSE Devices)
- idevice (Interoperable devices)
- ledge (SmartLSM VPN-1 Edge X gateways)
- lip40 (SmartLSM IP40 gateways)

For the last 2 type of objects, please see the notes at the end of this document section.

The object type is a **Required parameter**.

Example:
```
ofiller -t host -s 10.2.3.4 -d 10.2.3.99 -m 24 -o output.dbedit
```

-s - Source ip - Indicates the first IP we'll use to build the ranges. Note that when building SmartLSM Edge X or IP40 gateways, this initial IP cannot be smaller than 0.0.0.10 (and it's recommended to be 0.0.0.10). **Required parameter**.

Example:
```
ofiller -s 10.2.3.4 -t net -d 10.2.30.99 -m 24 -o output.dbedit
```

-d - Destination ip - Indicates the IP where the range finishes. It must be "bigger" (network-wise) than Source IP. Note than when building SmartLSM Edge X or IP40 gateways, this ending IP cannot be bigger than 0.0.254.254. **Required parameter**.

Example:

```
ofiller -d 10.2.30.99 -t net -s 10.2.3.4 -m 24 -o output.dbedit
```

-m - Mask - The mask that we'll use to build the objects. Must be between 8 and 30 bits. Required parameter.

00 bits = 0.0.0.0	08 bits = 255.0.0.0
09 bits = 255.128.0.0	10 bits = 255.192.0.0
11 bits = 255.224.0.0	12 bits = 255.240.0.0
13 bits = 255.248.0.0	14 bits = 255.252.0.0
15 bits = 255.254.0.0	16 bits = 255.255.0.0
17 bits = 255.255.128.0	18 bits = 255.255.192.0
19 bits = 255.255.224.0	20 bits = 255.255.240.0
21 bits = 255.255.248.0	22 bits = 255.255.252.0
23 bits = 255.255.254.0	24 bits = 255.255.255.0
25 bits = 255.255.255.128	26 bits = 255.255.255.192
27 bits = 255.255.255.224	28 bits = 255.255.255.240
29 bits = 255.255.255.248	30 bits = 255.255.255.252
32 bits = 255.255.255.255	

Example:

```
ofiller -m 25 -t net -s 10.2.3.0 -d 10.2.30.0 -o net.dbedit
```

-c - Color – The color we'll use to build the objects. See Appendix B for a list of valid Colors. **Optional parameter.**

Example:

```
ofiller -c sienna -m 25 -t net -s 10.2.3.0 -d 10.2.30.0 -o net.dbedit
```

-n - NAT ip - The IP behind which the objects will be automatically NATted. If not specified, no NAT will be done to created objects. Only Hide NAT is supported on this syntax. It cannot be used with -ns, -nd and -nm switches. **Optional parameter**.

Example:

```
ofiller -n 192.168.1.3 -m 25 -t net -s 10.2.3.0 -d 10.2.9.0 -o n.txt
```

-ns, nd, nm - NAT range Starting ip, NAT range Destination ip, NAT range Mask - The IP address range behind which the created objects will be Hide NATted. Sometimes there is a big network (let's say a Class B network) with internal addressing, and then a C Class network with valid addresses. These switches allow the administrator to bind every created object to a different IP from a declared valid network. This way, if we have the 10.10.0.0/16 invalid network, and then the 172.16.200.0/24 valid segment, we can use Object Filler to automatically create objects like 10.10.0.0/24 NATted behind 172.16.200.1, then 10.10.1.0/24 NATted behind 172.16.200.2, next 10.10.2.0/24 NATted behind 172.16.200.3 and so on. When 172.16.200.254 (the lastIP of the valid range) is reached, then the next object will be NATted using 172.16.200.1 (the first IP in the NATting range) again.. – **Optional parameters**

Example:

```
ofiller -ns 192.168.200.0 -nd 192.168.201.255 -nm 24 -s 10.10.0.0 -d
10.20.255.255 -m 24 -t net -o nets.dbedit
```

-b - hiding oBject - The name of the Check Point gateway object which will NAT hide the created objects. Optional parameter, but when specified -n must be also used. This object must exist on the SmartCenter before you attempt to use this switch, and the name of the object in the SmartCenter must be exactly as specified here, as all involved programs (including ofiller and dbedit) are case sensitive. If -n was given, but -b not specified, then objects will hide behind All gateways (*All) as default. **Optional parameter.**

Example:

```
   ofiller -b The_Wall -n 10.9.8.7 -m 25 -t net -s 10.2.3.0 -d 10.20.9.0 -o
x.txt
```

-o - Output file - The name of the file where resulting dbedit commands will be stored. Please make sure you have enough disk space to store all produced commands. To calculate this pace, take an average of 750 bytes per object to process. The File must not exist previously.. If it exists, the execution of the program will be aborted. This switch is mutually exclusive with -a (i.e. if you can only specify -a or -o, but must at least use one of them). **Required parameter if –a was not specified.**

Example:
```
ofiller -o dbedit_commands.txt -s 10.2.3.0 -d 10.2.3.9 -m 25 -t host
```

-a - Ascii file - The name of the file where resulting CSV information will be stored. This is an alternative to -a, and instead of writing dbedit commands, Object Filler writes information regarding the created objects (name, ip, comments, etc.) on CSV format, so you can take a look on a spreadsheet program first. Please make sure you have enough disk space to store all produced commands. To calculate this space, take an average of 120 bytes per object to process. File must not exist previously. This switch is mutually exclusive with -o (i.e. if you can only specify -a or -o, bust must at least use one of them). **Required parameter if –o was not specified.**

Example:
```
ofiller -a preview.csv -s 10.2.3.0 -d 10.2.3.9 -m 25 -t host
```

-v - Verbose mode - shows on the console (the screen) details on how the processing is being done line-by-line. This is very useful especially when importing files, since it says how each line was treated. **Optional parameter.**

Example:
```
ofiller –v -o dbedit_commands.txt -s 10.2.3.0 -d 10.2.3.9 -m 25 -t host
```

Building SmartLSM Objects with command line

When building SmartLSM ROBO gateways (ledge or lip40 object types), there are some rules that apply, and you must know:
- Object Filler assumes there are no previously created SmartLSM or regular Edge/Embedded NG objects previously created, nor any kind of profiles, nor any dynamic objects
- The first IP cannot be lower than 0.0.0.10
- The last IP cannot be higher than 0.0.254.254
- Automatically creates a SmartLSM VPN-1 Edge/Embedded NG profile called gen_profile with IP 0.0.0.8, used as the default profile on the created ROBO gateways.
- Automatically creates a Dynamic Object called gen_dyn_obj with IP 0.0.0.9 used on the created ROBO gateways.
- Automatically assigns the IP Address range 1.2.3.4-1.2.3.5 to all created objects.
- It doesn't assign any registration key for the created ROBO gateways.

Important note regarding IP addresses for SmartLSM related objects from command line
When building SmartLSM ROBO gateways with the command line, specified IP Addresses for the objects must be in the range from 0.0.0.10 to 0.0.254.254. Please note that you *must make sure* that no duplicate IPaddress exist on the configuration. To assure this, please log in to the SmartLSM GUI, sort the elements by "ID" (second column from left to right) and make sure the IPs you are specifying are not listed there. Then, use Object Dumper to dump the contents of your current Objects_5_0.C file and see that no profile or dynamic object is already using the IPs you're trying to assign.

To avoid any problems or IP conflicts in any case, is **highly** recommended to have the SmartCenter Server clean of SmarLSM objects, dynamic objects, profiles or any kind of dynamic objects. This is, you should use Object Filler just for the initial configuration, unless you know what you're doing.

The usual recommendation when building SmartLSM VPN-1/Embedded ROBO gateways using the command line, is to direct the output to a CSV file using the -a option of Object Filler:

```
ofiller -t lip40 -s 0.0.0.10 -d 0.0.0.210 -m 24 -a output.csv
```

Then edit the resulting file (output.csv on this case) to fill it with the proper ROBO gateway information. Finally, use the -i csv option to build the dbedit commands:

```
ofiller -f output.csv -i csv -o robogws.dbedit
```

This way, you can greatly automate the building of the new ROBO gateways.

Examples of program execution

```
ofiller -f source.csv -i csv -o objects.txt
```
Will take data from file source.csv , with CSV format and leave results (dbedit commands) on a file named objects.txt

```
ofiller -i csv -f source.csv -o objects.txt -v > results.txt
```
Same as above, but now the program's verbose output will be directed to a file named *results.txt*, instead of the console, so you may review it later.

```
ofiller -t hosts -s 10.0.0.0 -d 10.100.0.0 -m 24 -o hosts.dbedit
```
Will build hosts from 10.0.0.0 to 10.100.0.0 skipping network addresses as well as broadcast addresses, using 24 bits as objects netmask. Output will be directed to hosts.dbedit file

```
ofiller -t nets -s 2.0.0.0 -d 2.5.6.2 -m 24 -c blue -n 1.2.3.4 -b FireWall -o
nets.txt
```
Will build networks from 2.0.0.0 to 2.5.6.2 skipping broadcast addresses, using 24 bits as objects' netmask. Objects will be created on color blue, Hide NATed behind 1.2.3.4 and an object named FireWall. This firewall gateway name must be exactly as specified here, as program is case sensitive.

```
ofiller -f 535.conf -o imp_pix.csv -i pix -v
```
Will import a PIX configuration from a file named 535.conf and leave the output (CSV formatted) in the file imp_pix.csv

Importing configurations from files with Object Filler

Importing files in general

First thing you have to know is that Object Filler and Object Dumper don't need dos2unix conversions on ***input files***. This is, if you get a file from a Solaris or Nokia IPSO machine, you can get it to Windows, and it doesn't matter the format you transfer it with, will be processed the right way. However, for the ***output files*** may need dos2unix conversions if you move files to a different machine from which they were generated. If you're transferring a dbedit command file (the results from running Object Filler) over FTP, you must transfer it as ASCII, not Binary file.

Always try to use the ASCII output mode (-a switch) to review what Object Filler whould do, review the results, and then run Object Filler again on the original file, but using the dbedit mode (-o switch) to finally produce the dbedit commands you'll use to import into the SmartCenter.

If when you run Object Filler you don't choose the right file type (i.e. you ask to translate a PIX configuration, whereas you have in the file a NetScreen configuration), Object Filler will try to figure that out and will tell this suspicious status, but don't rely on this mechanism and always try to specify the right type of file.

Please be aware that if you have already an object that has the same name of an object you're importing, the only property of the object that will be modified is the IP Address, and if the object is also NATted, the NATting properties will be modified too...

In general, if you are going to populate a SmartCenter that already has data on it, it is strongly recommended to export your current list of objects, and compare it to the one that will be imported, may be using the ASCII output mode (-a) of Object Filler, and using Object Dumper to export your current configuration. This way you will notice which object have chances to be modified before you do any changes to your live configuration.

If possible, it is recommended to use Object Filler only to populate empty SmartCenter Servers.

Comma Separated Values (CSV) file type

This is by far the most powerful (but also the more complex) file format supported by Object Filler.

File must be formatted on CSV format, i.e. all values must be separated by a comma. The only special consideration is that columns order must be preserved as declared on the sample file and as is explained below. Non-used spaces can be just left empty (or filled with zeroes), but the space still has to be defined by a comma however.

Format of the CSV File used by Object Filler

This is the definition of the CSV file format used by Object Filler to take information to build or modify objects from. Also, it is the format on which Object Dumper leaves the information after processing the input files specified for it.

- Column 1 – *Object Name*
The name the object will have. Please consider the naming conventions for objects on SmartCenter. Usually the more important things to remember here are: No spaces are allowed (use dashes and underscores instead), names must start with a letter (no numbers), and limit the names shorter than 32 characters.

- Column 2 - *Type of object* or *operation*.
Use the same as the supported object types on Object Filler with command line, or the ones listed in the table with supported types above: *ss*, *cpfw, dynamicgw, cluster, member, host, net, plaingw, idevice, ose, range, dynamic, domain, interface, lprofile, ledge, lip40, tcp, udp, icmp, other, rpc, dcerpc*. Please see documentation below for building groups.

When creating interfaces, it's important that the interface is defined after the gateway that owns such interface is define. If you define it before, the creation of such interface will not complain on Object Filler, but at the import time with dbedit, it will fail.

If you are changing the properties of an object, it's also accepted on this column to have specified *modss, modcpgw, moddynamicgw, modcluster, modhost, modnet, modplaingw, modidevice, modose, modrange, modtcp, modudp.*

The type field can also be used to specify an operation. Currently RENAME (to change the name of a network object) and DELETE (to delete a network object) operations are supported. Provided object names must match the case of the real object names. Objects are not verified that they were processed by Object Filler before, nor that they currently exist on the SmartCenter.

- Column 3 - *IP Address*, *Initial IP Address*, *Port Number*.
This column usually contains the object's main IP address in "dotted" format like 1.2.3.4 – In the case of IP Address ranges, this column contains the initial IP of the range. In the case of TCP or UDP Services, it contains the port number, which can have a preceding > or < sign.

- Column 4 – *Netmask, Final IP Address, Timeout*
This column regularly specified the netmask in "dotted" format like 255.255.0.0 or 255.255.252.0. However, in the case of IP Address ranges, this column contains the final IP address of the range. In the case of TCP or UDP services, it contains the timeout for the service, which can be either "default" (the default global timeout specified in the SmartCenter), or a number in seconds.

- Column 5 – *Color* or *Owner device*
This column contains the color of the object. For a list of valid colors, please see Appendix B. When no valid color is specified, then black is assumed.
Howerver, if the object being specified is an interface, then this column contains the name of the object that owns such defined interface

- Column 6 – *NATting IP*, *Interface location*, *replies accepted*.
This column contains the NATting IP behind which the object will be NATted. This is optional.
When the object being processed is an interface, however, this column contains the interface location (internal or external). If the object is a service in the other hand, then it specifies if the service accepts replies or not. If they are accepted, then the column should list the word *replies*

- Column 7 – *NATting object*, *Interface Topology*
The name of the Check Point Gateway behind which the object hides. This is especially useful when the same SmartCenter is managing several Check Point firewalled gateways, and you want to perform NAT using only one of them. If not specified (if the column is empty, but NATting IP has being specified), or if *All* is used, then it will hide behind **All* the gateways managed by the SmartCenter.
In the case of Interfaces, this column specifies the IP addresses behind this interface (the topology). The valid values are *undefined*, which means there is no topology defined; *local,* which means all the IP addresses in the network specified by the interface's IP address; or the name of a network object (this has to be of type *network* or *network object group*) to be defined as the specific topology information for this interface.

- Column 8 - NAT type
It's the type of NAT that will be used for this object. Accepted values for this column are *Static* and *Hide*. If empty, but a NATting IP has been defined, Hide NAT type will be used by default.

- Column 9 - *Comments*
This column is used by all objects to put comments.

- Column 10 – Additional properties (*webserver*, *encdomain*)
When processing plain hosts, this column may contain the keyword *webserver*, which means that this object will be marked as a webserver for the effects of SmartDefense settings.
When processing Check Point Gateways, Check Point Dynamic Gateways, Check Point Clusters, Plain Gateways or Interoperable Devices, this column may contain the keywork *encdomain,* to specify that a manually defined encryption domain will be defined for this object

- Column 11 – *Protecting gateways*, *Encryption domain*
When processing plain hosts, and the host has been defined as webserver by the previous column, this column may specify behind which Check Point gateways this webserver is protected. If *All* is specified, it will be enforced behind all the gateways. If a gateway name is specified, this gateway will be the one specified as the protecting one. If leaved blank but webserver was specified in column 10, *All* will be assumed
When processing Check Point Gateways, Check Point Dynamic Gateways, Check Point Clusters, Plain Gateways or Interoperable Devices, this column contains the object (network or network object plain group) that will be used as the encryption domain, if the keyword *encdomain* was specified in column 10

Following are a couple of examples:

```
MyServer, host, 1.1.2.8,255.255.255.255, blue,          ,     ,     , HTTP Srvr
Users,    net,  1.2.0.0,255.255.0.0,     green,10.1.1.1, FW_3, Hide, users net
GW1,      cpgw,10.3.3.1,255.255.255.255,  black,        ,     ,      ,Main FW
eth0,     interface,10.3.3.1,255.255.255.0, GW1, external,    ,      ,
eth0,     interface,1.1.2.1,255.255.255.0, GW1, internal,Users,      ,
```

The first line will create a host named "MyServer" with IP 1.1.2.8, color blue and will have "HTTP Srvr" as comment.

The second line will create a network named "Users" with IP 1.2.0.0 and netmask 255.255.0.0 which will be Hide NATted behind the IP 10.1.1.1 and the Check Point gateway FW_3. In the comments field we'll have "users net" as comment. Object will be of color green.

The third, fourth and fifth lines define a Check Point Gateway with interfaces that belong to it.

When defining IP address ranges, you must define two IPs: the startingIP on the "IP address" column, and the ending IP of the range on the Netmask column (column 4). The ending IP must be "greater" network wise than starting IP, or Object Filler will reject it. The following is a valid example:

```
Int_Srvrs, range, 1.2.3.50, 1.2.3.60, green, 10.1.1.1, GatewayA, Hide, servers
```

This line will build an IP address range objects named "Int_Srvrs" from the IP 1.2.3.50 to IP 1.2.3.60, with green color, NAT Hide behind IP 10.1.1.1 and behind Check Point Gateway (which must be previously defined) GatewayA, and will use "servers" for the comment field.
When modifying properties, you may use the *mod* object types:

```
MyServer, modhost,1.1.2.8,255.255.255.255,blue , , , ,HTTP Srvr,websever,GW1
GW1,      modcpgw,1.1.2.1,255.255.255.255,black, , , ,Main FW,  encdomain,users
```

The lines above will modify a currently existing MyServer host object, will mark it as webserver for SmartDefense purposes, protected by gateway GW1
The second like will modify the already existing GW1 Check Point gateway, and will define the network *users* as the encryption domain for it.

CSV file type and Service objects

Since Object Filler 1.9.2 you can define TCP and UDP services using CSV files. Since Object Filler 2.0 ICMP, RPC, DCE-RCP and Other Services are also supported.

The format you must follow is this: *name, type, number, timeout, color, replies, expression*

name is the name you will give to the service.
type can be *tcp, udp, icmp, other, rpc* or *dce-rpc*
number is the port number that will be assigned to the service in the case or TCP and UDP services. This can be a single number, the indication ">" (as in >82) meaning whatever port bigger than the number specified, the indication "<" which means whatever port lower than the number specificed (as in <81) and also a range (as in 84-98) which means whatever port in between those 2. The rest of the columns is ignored. For DCE-RPC services, this column should contain the UUID of the service. For RCP services this column indicates the program number. For ICMP Services this specifies the ICMP type. For services of type Other, this column is the protocol number of the service.
Timeout is the timeout for the service (the time after which a session of this service would be considered no longer valid) in seconds. If *default* is specified, then it means that the default timeout specified for all the services of this type in the Global Properties of the SmartCenter, will be applied.

The following is an example of services definition:

```
udp_81,             upd,   81,    default,green,
tcp_bigger_82,      tcp,   >82,   600,
udp_lower_83,       udp,   <81,   default,
tcp_range_84-85,    tcp,   84-85, 1200,   blue,
sample_dcerpc,      dcerpc,12345678-90ab-cdef-0123-4567890abcde,,red
sample_other,       other, 87,    default,red,replies,ip_cmd=RIPCMD_RESPONSE
sample_rpc,         rpc,   100006,    ,   red,
sample_icmp,        icmp,  6,         ,   red,
```

CSV file type and Cluster related objects

Beginning with Object Filler 1.8, cluster objects are also supported with CSV files. To define clusters, there are 2 relevant object types, and the syntax is a bit different: First you've to define the Cluster Members (one line per cluster member), and then you have to define the cluster object itself. However, when defining the cluster object, you have to split it on several lines, indicating instead of the network mask, an object member that belongs to such cluster. All the other parameters have to be the same. The following is an example:

```
clmember1,  member,   10.2.99.1,  255.255.255.255,  blue, , , Cluster Member A
clmember2,  member,   10.2.99.2,  255.255.255.255,  blue, , , Cluster Member B
clmember3,  member,   10.2.99.3,  255.255.255.255,  blue, , , Cluster Member C
clusterA,   cluster,  10.1.2.33,  clmember1,       green, , ,Cluster Object
clusterA,   cluster,  10.1.2.33,  clmember2,       green, , ,Cluster Object
clusterA,   cluster,  10.1.2.33,  clmember3,       green, , ,Cluster Object
```

The first three lines define the cluster members. The last three lines define the cluster itself, and acknowledges clmember1, clmember2 and clmember3 as members of defined cluster "clusterA". Please note that all the fields are the same for the cluster, with the exception of the column to indicate the member.

Please also note than no other Cluster's properties are set (such as synchronization network or cluster topology), so this cannot be used to backup cluster configurations.

CSV file type and Groups

As of Object Filler 1.6, defining simple groups for network objects with CSV files is supported. Since Object Filler 2.0, defining service groups is also supported.

To do this, you must specify the name of the group on the name column (Column 1), the word "group" for network objects groups or "srvgroup" for service groups on the type column (Column 2), and then specify the name of the member on the IP Address column (Column 3). If the member name is not an object that was processed by Object Filler in this file (or predefined), it will reject this member. This behavior is by design, so the user knows that it's trying to include a member that was not created by the file. Following is an example:

```
group1,       group, Int_Srvrs , , , , , ,
group1,       group, Users     , , , , , ,
group1,       group, MyServer  , , , , , ,

tcps1, srvgroup, tcp_81  , , , , , ,
tcps1, srvgroup, tcp_gt90, , , , , ,
tcps1, srvgroup, tcp_lt20, , , , , ,
```

The previous lines will create an Object Group named "group1", whose members will be the previously created objects Int_Srvrs, Users and MyServer. Those lines will create a service group named "tcps1" whose members will be tcp_81, tcp_gt90 and tcp_lt20.

Colors and comments on groups is not supported.

Order is important when you are trying to add groups inside groups. If this is the case, make sure you have created the group you are trying to include as a member inside of another group...

CSV file type and Operations over objects

When using a CSV file as input for Object Filler, some operations over objects are supported. Currently only RENAME and delete operations are supported.

When specifying the RENAME operation, the first object name is the original one, and the last name (the one specified in the Column 3, where usually the IP Address of an object is specified) is thenew one for the object. RENAME doesn't change any object property, such as certificate's FQDN, just the name of the object.

When specifying the DELETE operation, the object name declared on the first column is the one that will be deleted.

The following lines are a sample of operation statements using a CSV files:

```
object1, rename, ObjectA  , , , , , ,
object3, delete,          , , , , , ,
```

The first line will produce the command to rename object1 to ObjectA. The second line will produce the command to delete object3.

Both operations are supported only over network objects. Such operations are not supported by Object Filler on service objects.

CSV file type is the only one that accepts operations over objects.

CSV file type and SmartLSM related objects

Since Object Filler 1.8, defining some SmartLSM related objects is supported. These objects include Dynamic Objects, SmartLSM VPN-1 Edge/Embedded profiles and SmartLSM VPN-1 Edge/Embedded ROBO gateways (types IP40 and VPN-1 Edge X Series). For this to work properly, SmartLSM must be enabled first on the SmartCenter, using "LSMenabler on" command.

The columns here are a bit different, and mean the following:

*** For Dynamic Objects**

- Column 1- name: The name of the dynamic object. Mandatory field.
- Column 2 type: Must be set to "dynamic" (Without the quotes). Mandatory field.
- Column 3 IP Address: IP for this Dynamic object. Please see note below regarding IP addresses for this kind of objects. Mandatory field.
- Column 4 Mask: Must be set to 255.255.255.255 - Mandatory field.
- Column 5 Color: Color for the object. Optional field.

- Columns 6, 7 and 8 are not relevant
- Column 9 Comment: The comment for the object. Optional field.

*** For SmartLSM VPN-1 Edge/Embedded profiles**
- Column 1- name: The name of the profile. Mandatory field.
- Column 2 - type: Must be set to "lprofile" (Without the quotes). Mandatory field
- Column 3- IP Address: IP for this profile object. Please see note below regarding IP addresses for this kind of objects. Mandatory field.
- Column 4- Mask: Must be set to 255.255.255.255. Mandatory field.
- Column 5- Color: Color for the object. Optional field.
- Columns 6, 7 and 8 are not relevant
- Column 9- Comment: The comment for the object. Optional field.

*** For SmartLSM VPN-1 Edge/Embedded ROBO gateways**
- Column 1- name: The name of the ROBO gateway. Mandatory field.
- Column 2- type: Must be set to either "ledge" or "lip40" (without the quotes). ledge means type set to VPN-1 Edge X Series. lip40 means type set to Nokia IP40. Mandatory field.
- Column 3- IP Address: IP for this profile object. Please see note below regarding IP addresses for this kind of objects. Mandatory field.
- Column 4- Profile name: It's the name of a SmartLSM VPN-1 Edge/Embedded profile previously created on this CSV file, or already existing on the SmartCenter. Mandatory field.
- Column 5- Dynamic Object: It's the name of a Dynamic Object previously created or already existing on the SmartCenter. It's mandatory only if you wish to assign an IP or range of IPs to be set as VPN domain behind the created ROBO gateway.
- Column 6- IP or Range of IPs: Only needed and processed if a valid dynamic object has been specified. The IP or range of IPs will be assigned to the dynamic object. If a range needs to be specified, then a dash (hyphen) must be used as a delimiter between the first and the last IP of the given range.
- Column 7- vpn/novpn: If set to "vpn", the previously assigned IPs to the dynamic object, will be exported as part of the VPN topology of this ROBO gateway. Optional field.
- Column 8- Registration key: If specified, this will be set as the Registration Key for this ROBO gateway. Optional field.
- Column 9- Comment: The comments for this ROBO gateway. Optional field.

To illustrate this, the following are some examples:

```
obj_dyn, dynamic ,0.0.0.19,255.255.255.255,blue,          ,     ,     ,Comments
prof1,   lprofile,0.0.0.18,255.255.255.255,blue,          ,     ,     ,Comments
edge_gw ,ledge,   0.0.0.20,prof1,obj_dyn,    192.168.10.40,vpn  ,pass1,Comments
edge_gw2,lip40,   0.0.0.21,prof1,obj_dyn,1.2.3.4-1.2.3.9,  novpn,word2,Comments
```

The first line just creates a dynamic object named "obj_dyn". The IP address specified is necessary. Please see note below regarding IP Addresses. The object is created on color blue and takes "Comments" as the comment for this object.

The second line creates a SmartLSM VPN-1 Edge/Embedded profile. Again, the IP address is necessary. Color for the object is blue and the comments are simply "Comments".

The third line creates a SmartLSM VPN-1 Edge/Embedded ROBO gateway with type of it set to VPN-1 Edge X Series. The IP is needed (please see note below regarding IP addresses). Then uses "prof1" as profile (this SmartLSM VPN-1 Edge/Embedded profile must exist previously on the SmartCenter, or must be created previously on the same CSV file), uses "obj_dyn" as the Dynamic Object, and assigns the IP 192.168.10.40 as the value for this dynamic object. Also specifies that this Dynamic Object belongs to the VPN domain, and sets the Registration Key to "pass1".

The fourth line creates a SmartLSM VPN-1 Edge/Embedded ROBO gateway with type set to Nokia IP40. The profile is "prof1", the Dynamic object is "obj_dyn", but this time the range

assigned to the dynamic object is from 192.168.20.10 to 192.168.20.40 (i.e. a range instead of a single IP). Please note the hyphen (dash) separating both IP addresses. The Registration Key is set to "word2".

Important note regarding IP Addresses for SmartLSM related objects
When building SmartLSM ROBO gateways with the command line, specified IP Addresses for the objects must be in the range from 0.0.0.10 to 0.0.254.254. Please note that you *must make sure* that no duplicate IPaddress exist on the configuration. To assure this, please log in to the SmartLSM GUI, sort the elements by "ID" (second column from left to right) and make sure the IPs you are specifying are not listed there. Then, use Object Dumper to dump the contents of your current Objects_5_0.C file and see that no profile or dynamic object is already using the IPs you're trying to assign.

To avoid any problems or IP conflicts in any case, is **highly** recommended to have the SmartCenter Server clean of SmarLSM objects, dynamic objects, profiles or any kind of dynamic objects. This is, you should use Object Filler just for the initial configuration, unless you know what you're doing.

CSV file type and importing security rules

Since Object Filler 1.9.2, importing basic security rules from CSV files is possible. When importing security rules, the syntax for the line is the following:

```
security_rule,source,destination,vpn,service,action,track,install_on,time,
comment
```

Where:
security_rule is a key word that specified object filler to treat this line as a security rule definition. It must be like this.
source is a network object. You can specify several, using a semicolon (;) as separator. Currently, having user groups as source (for authentication rules or VPN rules) is NOT supported.
destination is a network object. You can specify several, using a semicolon (;) as separator.
vpn is the VPN communities for this rule
service is the service object for this rule. You can specify several, using a semicolon (;) as separator. Currently using resources (uri, smtp, etc.) is NOT supported.
Action can be accept, log or drop. Any other action is NOT supported
Track can be Log or None. Any other action is NOT supported.
Install_on can be any Check Point gateway object, or the word "Any". You can specify several, using a semicolon (;) as separator.
Time can be a time object, or the word "Any". You can specify several, using a semicolon (;) as separator.

The following are examples of valid rules defined:

```
security_rule, Server1,      Srv2,      Any, tcp_81,     accept, log, Any, Any,
security_rule, Host_X;HostY, Any,       Any, http,       accept, log, Any, Any,    XYZ
security_rule, Internal_LAN, Srv1;Srv2, Any, NBT,        accept, log, Any, Any,    Comment
security_rule, LocalMachine, Any,       Any, icmp-proto, drop,   None,Any, Any,
security_rule, InternalNet,  Any,       Any, ftp;telnet, accept, Log, Any, Any,
security_rule, Any,          Any,       Any, Any,        drop,   log, Any, Any,
```

The processing of rules is affected by the Object Filler switch –nopv – If this switch is not specified, Object Filler will try to check that the objects specified in the rules were processed before (or are part of the predefined objects). If they were not processed (or predefined), they will be substituted by "Any".

If –nopv is specified, the checks mentioned above are not performed.

Section headers are properly recognized and processed both by Object Dumper, and Object Filler while importing rules from CSV Files. If a section header is specified, it should be done using the keyword *section_header* in the first column, instead of *security_rule*, as in the following example:

```
section_header, OPSEC_rulebase
```

List (list) file type

Specified file must contain 2 mandatory fields: IP Address and netmask. Additional optional columns are color, IP behind which NAT will be done, object behind which NAT will be done, and NAT type (Hide or Static). The explanation for all those columns is exactly the same as for the CSV file type.

Object Filler automatically calculates (based on provided netmask) if the object is a network, a host, an IP Address range or a group, then generates a name (unless it's a Group, for which the name it's expected to be the first parameter) and the appropriate network object. Due this, the only supported object types are hosts, networks, IP address ranges, and groups. Check Point Hosts, Check Point Gateways, Check Point Dynamic Gateways, Plain Gateways, Interoperable Devices, OSE devices and the others are not supported on this type of file. If you need those, please take a look on the CSV file type.

When building IP Address Ranges, you must enter the starting range IP on the first (IP Address column) and the ending range IP on the second (netmask) column. Ending IP must be "greater" network-wise than the starting IP.

If you whish to build Groups, all you have to do is to specify the name of the member on the IP Address column, and the IP of the member on the netmask column. If the member (i.e. if the object corresponding to this IP) was not processed before in this file, Object Filler will reject this member. This behavior is by design, so the user knows that it's trying to include a member that was not created by the file. You cannot include groups as members of another group while you are importing a List type of file: this is supported only with CSV files. Also, comments and color for groups is not supported.

Hosts (hosts) file type

Indicated file has the format of a hosts file (/etc/hosts on Un*x systems, or %SYSTEMROOT%\system32\drivers\etc\hosts on Microsoft Windows systems). Object Filler automatically generates hosts objects using the name and IP listed on the file.

When importing hosts files, you may specify an object type besides plain hosts in the Object Filler's command line with the –t switch, so you can actually build OSE Devices or Plain Gateways for example.

Cisco PIX (pix) file type

When importing from Cisco PIX, following versions were tested: 5.1(1), 5.1(2), 5.1(4), 6.1(4), 6.2(2), 6.3(1).

The file entered as input for this option is the configuration listing from a Cisco PIX device. You can get this information from a PIX device using the command "show running" or "write terminal".

When importing from Cisco PIX, Object Filler will only recognize plain Hosts, OSE devices (for the interfaces of the PIX device itself) and Networks. Object Filler will recognize all valid IPs that are listed in the configuration, not only those from the rulebase, and will process it.

Names on the objects are assigned according to the object type recognized (OSE, Host or Network).

By default NATted objects (Static or Hide) are supported. As a matter of fact, Object Filler by default processes the NAT statements first

In the case of static NAT, NATting to the outside interface it's privileged, this is, if the same IP is NATted on several interfaces, Object Filler will try to leave as the imported NAT the one that faces to the outside interface. If no outside interface is declared, then the first static statement found is applied.

If several global statements are bound to the same NAT ID, only the first IP of all of them will be used, and the outside interface will be preferred also.

In the other hand, all NAT statements are processed. If several NAT statements belong to the same NAT ID, all of them are processed to the first global IP specified for such NAT ID, as explained before.

Object Filler won't process ranges (i.e. when IP addresses are in the format aaa.bbb.ccc.ddd-www.xxx.yyy.zzz - Example: 1.2.3.4-1.2.3.10). In those cases, the program will split the range and will take in account just the first IP (1.2.3.4 from our example).

NAT processing is also affected by the –nonat switch of Object Filler. If this switch is specified, no NAT processing will occur at all.

Since Object Filler 1.9.2 the import of rules from Cisco PIX configuration files is also supported. The only supported rules that may be imported are the ones specified with the *access-list* statement. To make this happen, you have to specify the –p (policy) switch in the Object Filler's command line.

If there are several access-lists in the same configuration, all the access-lists will be imported, but they will be separated using a standard policy tag in the imported configuration

To open the imported policy in the SmartDashboard (once you have imported the configuration via dbedit), go to File, Open. You will see the Object Filler imported policy there.

Juniper/NetScreen ScreenOS (netscreen)

When importing from NetScreen devices, ScreenOS from NS5XT, NS100, NS500 and NS5200 devices were used for testing. Tests have been conducted using ScreenOS 4.X and 5.X versions of the OS.

The file entered as input for this option should be the configuration listing from the NetScreen device. You can get this information from a NetScreen device using the command "get config all".

When importing from NetScreen, Object Filler will only recognize Hosts and Networks as valid types. Object Filler will recognize all valid IPs (not only those from the rulebase, but any IP) and process it. However, only Check Point gateways (for the IPs of the device itself), plain Hosts and Networks will be recognized and built.

Names on the objects are assigned according to the object type recognized (Check Point Gateway, Host or Network)

Static NATted (mip) objects are supported. Hide NATted (dip) and PAT (vip) objects are not supported.

SecureComputing Gauntlet (gauntlet)

Importing from Gauntlet was tested with version 5.5 running over Solaris. The configuration files needed may be found under /usr/local/etc/mgmt - but this may change.

Newer Gauntlet versions should work, but were not tested. Any reports of Object Filler running over other versions would be appreciated.

Only hosts and subnets are recognized, no other types of objects are built. Name is not imported from file. Instead, a new name is built according to the object type recognized.

No NAT conversions are done while converting from Gauntlet, mainly because of the lack of more testing files.

If you need to convert from Gauntlet and have some problems, have sample files willing to share, or have documentation of something unsupported on Gauntlet that should be here (like Groups, NAT support), please send me an e-mail.

SecureComputing SideWinder (sidewinder)

When importing from SideWinder, version 5.21 patch 9 configuration files were used for testings, with the contents of both ACL and IPFilter settings.

When reading the ACL configuration the following tables are supported: ipaddresses, hosts, subnets, and netgroups. Domains and servicegroups are not supported yet.

When the object has a name (for hosts, subnets and netgroups), this name is kept on the build object. If object has not a name, a name is created according to the object type recognized.

Hosts, networks and groups are properly recognized.

Object Filler also takes the IPs found on ACL or IPFilter statements. When importing the IPFilter statements, NATted IPs are converted properly, always using Hide NAT.

Symantec Raptor (raptor)

When importing from Raptor, version 6.03 for Windows was tested.

The file used is the gateway.cf, which contains the IPs and rules used for the configuration.

Hosts and networks properly recognized, as well as declared TCP and UDP services that are declared by port and have no name on them.

No NAT statements are supported on this version.

Cisco IOS Router (ciscorouter)

When importing from IOS configurations, versions 11.0, 11.2, 11.3.3.T, 12.0, 12.1 and 12.2 were tested.

Hosts and networks are properly recognized. No NAT statements are supported on this version.

If –p switch is used in Object Filler, and the configuration contains rules, the rulebases are processed accordingly.

Importing Object Filler's output to a Check Point SmartCenter Server or Provider-1 MDS Server

Modifying Object Filler's Output before importing

Since all output is directed to a text file, it's feasible to edit this file using any text editor, and modify (as an example) the prefix for the object's names (Net for Network, or Host for IP; as examples) or do any other modification you may need. This is true in both cases: for CSV formatted output (-a switch) and for dbedit commands output (-o switch).

Using dbedit to process Object Filler's results

First of all, it is greatly suggested you to read the following articles on the public partition of SecureKnowledge:

- https://secureknowledge.checkpoint.com/sk/public/idsearch.jsp?id=skI3301
  Editing the object_5_0.C file using the dbedit utility

- https://secureknowledge.checkpoint.com/sk/public/idsearch.jsp?id=sk10104
  Using the dbedit utility to modify the value of a specific network object property

- https://secureknowledge.checkpoint.com/sk/public/idsearch.jsp?id=sk12222
  Using queryDB_util to query the database

If you have access to the registered partition of SecureKnowledge, you may find the following articles also useful and interesting:

- Using dbedit utility to create network, host and group objects, and place network and host objects in group objects
  Solution ID: sk22957

- Creating Service Groups, Services, and Adding Services to Groups using DbEdit
  Solution ID: sk30370

- Using the dbedit utility to modify the value of a specific network object property
  Solution ID: sk10104

- Running command line 'dbedit' in a CMA environment
  Solution ID: sk23802

- Update command fails to execute properly using the dbedit utility
  Solution ID: sk10098

- Downloading and installing Check Point Database Tool utility
  Solution ID: sk13009

Then, it's important to remember that Object Filler's output files can be transferred from one machine to another. So it is **not** necessary to have Object Filler running on the same machine where the target SmartCenter Server is sitting. This SmartCenter Server could be in a different machine, and even a different operating system than the one used to run Object Filler.

If you are going to use the Object Filler's dbedit commands file result in a different machine from the one used to generate it, please verify that the proper dos2unix conversions (converting CR+LF to CR only) have been done, when you are passing files between machines with different operating systems (Windows to UN*X).

Keep in mind that dbedit commands are 100% ASCII text and should be treated accordingly when transferring using FTP-like mechanisms. If you start to see an error "Token contain illegal character" then you're probably transferring the file in the wrong format. Please verify that, if you're using FTP, you establish the transfer mode to "ASCII" instead of "binary" (which is the default sometimes). If you are transferring the files using diskettes, and the source and destination machines have different operating systems, dos2unix conversions may also apply.

Besides that, when importing files to the SmartCenter using dbedit, please make sure that:
- Your management processes are up and running. In the SmartCenter Server machine you can use the command "`cpstat mg`" or "`cpstat mg –h <IP address>`" to verify it.
- Your SMART Client (GUI clients), especially the SmartDashboard, are not running. If you strictly need to use them while importing, then please log in to the SmartCenter Server as read-only while you do the import.
- You are using a user with administrative privileges at operating system level (root, admin, Administrator or equivalent) If not, then change to a higher privileges user or a user that has enough permissions to run Check Point's binaries and affect Check Point's configuration.
- The IP from which you are running dbedit is declared as a valid Smart Client (GUI Client) IP. If not, then add it using cpconfig. In Provider-1 environments, you may need to also add the IP addresses for the MDS and/or the CMA itself as GUI Clients into the target CMA's configuration.

If you are running Provider-1, besides the above, also make sure that:
- You're doing the process on a MDS Manager or MDS Manager and Container server.
- You set the proper environment (using "mdsenv cma") before trying to connect using dbedit.
- You use the CMA's IP address as target for dbedit (dbedit's switch -s), and NOT the MDS IP Address.

You may try to run dbedit first and see if you can get into the target SmartCenter/CMA without any problems. Then you simply have to import the file using "-f" switch from the operating system command line, like in the following examples:

```
dbedit -f output_sample.txt
```

In the above case dbedit will read input from the file "output_sample.txt". This will prompt for the SmartCenter Server IP Address, an administrator username and the administrator password.

```
dbedit -s localhost -u admin -p duckystyle -f nat_networks.txt
```

In the above case dbedit will read input from the file *nat_networks.txt*, specifying that the SmartCenter Server is the located at the localhost, using *admin* as administrator's username and *duckystyle* as admin's password.

```
 dbedit -s 10.20.30.55 -u ccse -f nat_networks.txt
```

In this case dbedit will read input from the file *nat_networks.txt*, specifying that the SmartCenter Server or CMA is located at the machine with the IP 10.20.30.55 using *ccse* as administrator's username and asking interactively for the administrator's password.

If you get any error message or weird behavior while trying to import the objects you created with Object Filler, please consult Appendix A to see common causes of known problems.

Object Dumper

Program syntax:

```
odumper help  (prints help pages)
odumper -f file [-p file] -o file  [-d] [-html] [-v]
```

-f specifies the path to the objects (Objects_5_0.C or objects.C) file you want to process
-p specifies the path to the rulebases (rulebases_5_0.fws) file you want to process - Optional
-o specified the path to the output formatted file you want to have
-d tells the program to also print the default objects - Optional
-html formats the output to HTML (instead of default CSV format) - Optional
file is a valid filename - such as output.txt, output.html or objects.C

Required parameters: -f and -o

If you want to redirect the program's output, you can use the operating system ">" operand to do so.

Please note all parameters are case sensitive.

Program syntax #1 : Asking for help

```
odumper help
```

Prints every possible command line combination.

Program syntax #2 : Importing from an Objects_5_0.C, rulebases_5_0.fws and/or objects.C file

```
odumper -f file [-p file] -o file [-d] [-html] [-v]
```

Please note all parameters are case sensitive.

-f - input File - It can be an Objects_5_0.C file taken from the $FWDIR/conf (or %FWDIR%\conf) directory from a SmartCenter Server. It can also be an objects.C file taken from the $FWDIR/database (or %FWDIR%\database) from a Check Point Gateway (Enforcement Point) in a distributed configuration. Also you may use Check Point FireWall-1 4.1 objects.c files (located under $FWDIR/conf/objects.C) From this file the program reads the objects definitions, so they can be displayed after. **Required parameter**

Example:

```
odumper -f copy_of_Objects_5_0.C -o output.csv
```

-p - Policy File - It must be the rulebases_5_0.fws file, taken from the $FWDIR/conf (or %FWDIR%\conf) directory from a SmartCenter Server. From this file the program reads the rules definitions. **Optional parameter**.

Example:

```
odumper -f copy_of_Objects_5_0.C -p Copy_of_rulebases_5_0.fws -o output.csv
```

-o - Output file - The name of the file where resulting objects information will be stored. Please make sure you have enough disk space to store all produced information. To calculate this space, take an average of 150 bytes per object to process. **Required parameter**.

Example:

```
odumper -f copy_of_Objects_5_0.C -o output.csv
```

-html - HTML format for the output file - when this switch is specified, the output written to the file specified by the -o switch, is formatted on HTML using tables, and can be viewed by any standard web browser. Mozilla 1.7, Internet Explorer 6.0 and Netscape 7.2 for Windows were tested. **Optional parameter**.

Example:
```
odumper -f copy_of_Objects_5_0.C -o output.htm –html
```

-v - Verbose mode - shows in the console (the standard output, the screen) details on how the processing is being done line-by-line. This is very useful especially when debugging, but not in other circumstances since the output can be really overwhelming and a bit meaningless for most of the times. **Optional parameter**.

Example:
```
odumper -f copy_of_Objects_5_0.C -o output.csv –v
```

Modifying Object Dumper's Output and Importing Back

Since all output is directed to a text file, it's feasible to edit this file using any text editor, and modify anything there. However, due the format used (Comma Separated - CSV), it's more easy to edit files produced by Object Dumper using any spreadsheet program able to open CSV files, such as Microsoft Excel.

Files produced with Object Dumper, can be converted to dbedit files again, using Object Filler's CSV option (-i csv). Any modifications made to the file can be imported back to the SmartCenter this way.

Remember, that if you are modifying a configuration to import it back with Object Filler, you should change the object types accordingly: *modhost* instead of *host*, *modnet* instead of *net*, and so on. Please see the table of supported objects for modifications in the beginning of this document.

Web interface for Object Filler and Object Dumper

Both programs have a single, not fully featured and shared *proof of concept* web interface, which is provided here in two files:
- ofiller.html - Is the HTML code that acts as front-end for the user.
- ofiller.pl - Perl Code that processes as CGI module, all the data captured by the front-end.

To make this web interface usable, you must have Perl installed on your computer. In our case we tested using the Perl package provided by ActiveState found here: http://www.activestate.com/Products/ActivePerl/ when testing on Windows, and the Perl distribution provided with Red Hat Linux 7.2 while working on GNU/Linux. You need also to have a Webserver running. This was tested using Apache Web server 1.3.27 for Windows and for Linux, and also Internet Information Server (obviously under Windows).

You should place ofiller.html inside a public HTML folder, available for document publication from the webserver you are using. You should place ofiller.pl on the cgi-bin directory for such webserver.

It's also necessary to modify the following lines inside ofiller.pl:

* my $PATH_TO_EXEC
  This should reflect the path where Object Filler and Object Dumper executables are available (ofiller.exe and odumper.exe, or their GNU/Linux versions). It should not include the program names themselves, just the path.
  Examples:

```
my $PATH_TO_EXEC = "d:\\ofiller\\cgi-bin\\";
my $PATH_TO_EXEC = "/usr/local/ofiller/";
```

* my $OFILLER_EXE
  This should contain the name of the Object Filler executable. Examples:

```
my $OFILLER_EXE = "ofiller.exe";
my $OFILLER_EXE = "ofiller.lin";
```

* my $ODUMPER_EXE
  This should contain the name of the Object Dumper executable. Examples:

```
my $ODUMPER_EXE = "odumper.exe";
my $ODUMPER_EXE = "odumper.lin";
```

* my $UPLOAD_DIR
  This should point to an empty directory that must exist before the program can be run. It is used to upload and process configuration and Objects_5_0.C files that will be processed by those tools. Examples:

```
my $UPLOAD_DIR = "d:\\ofiller\\upload\\";
my $UPLOAD_DIR = "/usr/local/ofiller/upload/";
```

I would like to greatly thank Pedro Paixão, Check Point Latin America Regional Technical Consultant & SE Manager for his help on this web interface.

Appendix A. Frequently Asked Questions

Here is my try to condense questions that I get asked most of the time, in an attempt to provide fast and concise answers. Please feel free to write me an e-mail if you don't find an answer to yours. I promise I'll reply to it.

1. General Questions

Where can I get the latest versions?
Please consult the section "Programs Availabilty" above on this document.

Who maintains Object Filler and Object Dumper?
So far, Martín Hoz (mhoz@mexico.com)- Security Engineer for Northern Latinamerica at Check Point Software Technologies, is the person that maintains this (but this may change in the future), not without the valuable help of persons that assisted a lot with information on the Check Point products and also people that tested it version after version. "thanks".

Are these tools supported by Check Point or some other entity?
No. These tools are not officially supported by Check Point nor somebody else, in any way. Please use them at your own risk. Any good or bad result of using these tools either directly or directly is only responsibility of the person using them. Please read the Disclaimer included in this file for more information...

Got a problem, can I e-mail you?
Sure! - and I always answer my e-mails. But please, before doing so, read the provided documentation and make sure you're using the latest version of the tool, as I regularly audit and fix the code (i.e. try to make it support "can't happen" situations) while adding new features. The list of available sites to download the latest version of the tools is listed above.

What is the origin of Object Filler?
Pain on my fingers. Really. As a presales systems engineer, several times found myself on the duty of filling the SmartCenter Server with tons of objects which was a bit tiring, boring and painful doing it by clicking all the time. So I thought of more automatic way on doing it, and also trying to ease the process of importing configurations from other brands (something also needed every now and then - and more and more frequently in recent times, once people realize why Check Point is superior ;-). Given the open and robust nature of SmartCenter and the powerfulness of dbedit, this task was not hard at all... and seemed natural to be done…

What is the origin of Object Dumper?
Just to have another tool to dump the Objects_5_0.C and rulebases_5_0.fws on a more readable and easy to manipulate format, and also have a companion tool for Object Filler for exporting and importing back configurations.
Object Dumper is not intended to be a tool to document the configuration of your SmartCenter. If what you are looking for is a documentation tool, I strongly recommend you to take a look on a couple of good tools that do similar (even better in my opinion) job on this:

- Web Visualization Tool:
  http://www.checkpoint.com/techsupport/downloadsng/utilities.html#visualization
  Officially supported by Check Point, that supports exporting rules and objects to HTML/XML format.
- FW1Rules:
  http://www.wyae.de/software/fw1rules/
  Unsupported by Check Point tool, that allows to export both objects and rules in several formats, including HTML and CSV. Written on PERL.
- CPrules:
  http://www.wormnet.nl/cprules/
  Unsupported by Check Point tool that allows to export the rulebase to HTML. Written on PERL.

Object Dumper was not intended to be a backup/restore or migration tool either. Please check the Related Programs section in this document, as well as the Object Dumper section below, to find out more on this.

Why "Object Filler"? Why "Object Dumper"? - Where did you get those names?
Just names that came to my mind. I thought they were original and had a meaning for what the tools intend to do. Sorry if they look ugly to you. :-P

Are there any minimum requirements (processor, RAM, disk, etc.) to run these programs?
Not really. The program uses the disk to store the output, so just make sure you have enough disk to store this. Having 1 GB of free disk would be way more than enough and safe in most of the cases. In the other hand, the program itself plus the internal data would need around 32 MB on RAM (assuming you will process *thousands of objects*). Generally speaking, having these resources will be more than OK. One last word on memory though: On PCs with less than 128 MB of RAM I've been informed that sometimes the program just doesn't do nothing and stops with no error message or anything. I'm still trying to catch up where the bug is, so if this is your case, please let me know. Processor will just speed up calculations, but since both programs are real small and not intensive on processor usage, any processor is ok. Same goes to the bus.

Why did you prefer to parse text files and to use dbedit, instead of using the CPMI OPSEC API?
Because of three reasons: I like it better the way it works now, and it simply works (remember I'm doing this for fun! ;-). Then because it's clearer to people how it works and what the results are (so, it's easier to make people to trust in the tools, because they understand what is done and how is done), and allows them to change things if they want to; and finally because to me this was always and (still is) just a proof of concept, which is always easier to do with text. I never thought Object Filler & Object Dumper would be at the stage they're now... and I honestly never thought somebody else would use this except me and a couple of friends, so it was more like a personal toy & hobby, but there you go...

What development environment/programming language are you using to develop these?
I use standard ANSI C for programming. Why not PERL or something else? Because in general I don't like interpreted languages like PERL (except SHELL scripting). Also, because I don't know (and unfortunately don't have the time to learn) PERL. Then, because at this time I only remembered C programming, because I like C, and finally because I want it to be C. Right? ;-)

Generally speaking, I don't really use a development environment. While on Microsoft Windows, I use gvim (http://www.vim.org) and GCC 3.4.3 (I use the DJGPP flavor - http://www.delorie.com/djgpp/) – occasionally I also use the Visual C++ Toolkit 2003 (which includes a free command line compiler). While doing stuff on Solaris and GNU/Linux, I use an standard GNU vi and GCC 3.2.2

Are these tools available for free (at no cost), or do I need to pay something?
Yes, they are free if what you're asking is if they are available at no cost. They are not really free in the sense of freedom, since they really belong to Check Point Software Technologies as intellectual property (even if they are not officially supported) because they were written by a Check Point employee (me). However, if Object Filler or Object Dumper were useful for you, I would like to hear about it, so I'd ask you to "pay back" giving me feedback about your experience (especially in environments not documented as tested on)...

I really want to payback somehow with money or something like it... How can I do it?
I extensively use gvim (http://www.vim.org), a free text editor which asks in return to support poor children in Uganda through buying a book or making a donation (I already bought the book). I'd ask you to participate if you want to payback with money, either buying the book (if you like vi you will love to use gvim, which I recommend, and the book gives you interesting ideas, so it won't be a waste of money, and you will be helping) or donating something. Here is the link: http://iccf-holland.org/click5.html - In México we also have poor people (unfortunately, like everywhere, but here sometimes is also really bad), so you can also donate to the following organizations that somehow assist poor child and people in need in general, in México and other places also. :
http://www.mexico-child-link.org/
http://www.cruzrojamexicana.org/donativos/portarjeta.php
http://www.redcross.ca/
http://www.redcross.org/
http://www.unicef.org/
http://www.oxfam.org/
http://www.savethechildren.org/
http://www.msf.ca/

Remember that there's always somebody in this world that needs our help. You needed the assistance of this tool, and some people needs our money to have some food or better means on this life: I trust the tools saved you enough bucks or time to make you willing to give a (even small) donation to any of those organizations (or even better, all of them ;-) or in general, to any other organization that tries to make this world better somehow…

Finally, if you really want to send me a gift, I enjoy very much getting postcards from everywhere in the world, so if you have the time, please send me one... (but please, only if you already donated something to the charities above listed ;-) - As of February 2005, my current mail address is:

Martín Humberto Hoz Salvador
Querétaro 162 Depto. M-203 esq. San Bernabé
Col. Progreso Tizapán.
Del. Álvaro Obregón
CP 01080
Ciudad de México, Distrito Federal – MEXICO

Do you have the source code of the tools available for users to read it or modify it?
No. I can't distribute the source since it doesn't really belong to me, as I explained in a question above. Sorry.

How big is the source code for both tools?
So far, this is the wc (word count) report for the source code of the tools:
For Object Filler
20,110 lines; 62,061 words; 732,114 chars
For Object Dumper
5,147 lines; 14,289 words; 176,133 chars

Can I redistribute these utilities on my website/ftp site? Can I redistribute these utilities together with my package or software?
If you will charge for that in any way or lock the distribution in any way, the answer is definitely no. Otherwise the answer is a maybe. Send me an e-mail (address available at the Contacting the Author section) if you're planning and have the way to do so and we'll discuss it.

On what platforms can these utilities run?
Currently the platforms on which the tools run natively are Microsoft Windows, GNU/Linux (Red Hat Linux) and Check Point SecurePlatform. I was informed that it also runs on other Linux distributions like Mandrake or Suse, but not confirmed it by myself. I'm planning to add support for Solaris for next versions if there's some demand. However, remember that you can put and user the output of them wherever you want. I've done it with Nokia IPSO and Solaris itself for example.

On what Check Point versions were these tools tested while under development?
2.0 version of the tools was tested on Next Generation with Application Intelligence R55; mainly. Also was tested a bit on NG+AR R55W.

2. Object Filler

What's the best way to invoke Object Filler when importing files?
Use always the "-v" option, and then send the output to a file. Then review with a text editor such output file to see the details of what happened, and look for possible errors on the processing. The syntax should be something like this:

```
ofiller -f import.csv -i csv -o objects.dbedit -v > output.txt
```

Then edit it:
`notepad output.txt` (or "`vi output.txt`", or "`edit output.txt`", or whatever is needed according to your text editing preferences).
This way you will be able to see how the processing was done line-by-line.

I also strongly recommend exporting your current list of objects, and comparing it to the one that will be imported. This comparison may be accomplished using the ASCII output mode (-a) of Object Filler, and using Object Dumper to export your current configuration to CSV. This way you will notice which objects have chances to be modified, *before* you do any changes to your live configuration.

A good tool to compare text files that I like is CSDiff, which you can find here:
http://www.componentsoftware.com/products/csdiff/index.htm

Why building SmartLSM VPN-1 Edge/Embedded ROBO gateways or profiles, doesn't work in my SmartCenter?
Please make sure you have SmartLSM enabled on your SmartCenter. To do so, use `LSMenabler on` command from the Operating System command line, on the machine where your SmartCenter sits. Also, please keep in mind that this feature was developed and tested for NG+AI R55. If you're trying with a newer version and it doesn't work, please send me an e-mail.

Is it possible to modify the IPs of a massive number of objects (i.e. change all 192.168.x.x objects to 172.16.x.x) somehow using Object Filler?
Yes, in combination with Object Dumper. First, export your objects information with Object Dumper. Then, using any Spreadsheet or text editor program, edit the file and select the objects you are interested on changing. After that, using the search & replace facility of your editing program, change the IPs you want to change (in the example case, use "search '192.168.' and replace with '172.16.'"). Also replace the type of the object, with a preceeding "mod". For example, if the object's type is "host", replace it by "modhost". If the type is "net", replace it by "modnet" and so on. Then, import this information again using Object Filler (and CSV file option). This will do the trick. Don't worry about other property of the objects (like certificates), since the only property of the object that will be modified is the IP Address. If you would like to modify also other properties (Certificates, etc.) Object Filler can't help you on that.

Why Object Filler doesn't support importing rules from NetScreen, Gauntlet, SideWinder, etc.?
In previous versions, no rule importing was possible at all. This was mainly because of two big reasons:

- Because when migrating is a good chance to review the rulebase, so it's better to review what you're going to configure in your brand-new Check Point VPN-1 Pro/Express.
- Because doing rule translation between firewalls is not easy ;-) especially when the philosophy is different (example: proxy or packet filter, versus Stateful Inspection)

Since version 1.9.2 it supports importing basic rules from Cisco PIX and Cisco Routers, as well from CSV Files.

If you think Object Filler should support importing rules from other brands, please send me an e-mail (explaining your reasons too), and if I get enough requests, I'll try to do it! :-) – if you do so, please send me also example files of the brand you wish to support, as if I get more information on the file structure, my job will be easier and you will get results faster ;-). You may sanitize such file changing IP addresses or names, just please keep the file structure, so I can work on it.

Are you going to support importing configuration from X brand of firewall, or X type of file?
May be. When I released Object Filler1.2 I thought that it would be the last release ever. Then I got myself some other tasks that could be eased using Object Filler and then decided to increase the functionality, including new types of objects and other configuration files for other brands. So, if you have suggestions on what other file types Object Filler should support and/or you have sample configuration files for other firewall brands (you may sanitize them by changing names and IP addresses if you want, but if you do so please keep the file structure), or simply something you think can be eased with some extra functionality on the tools, please send me an e-mail.

Can I help on the process of supporting a new file type?
For sure! - if you have samples of the type of file you would like to support, you can submit the files to me. Please send me an e-mail with this information. I don't need the real names or IP address information (so you can use search & replace of notepad, vi or whatever, to substitute that and "protect the innocents"). Just please leave the format of the file intact, so I can analyze it correctly and find the proper pattern matching for it. And of course, if you allow me to, I'll mention your name on the thanks section. ;-)

What happens if while importing a configuration file, I choose the wrong type? (i.e. if I choose Cisco PIX when in fact it's a SideWinder configuration).
Object Filler will process it, but just not the right way. Usually NATs statements, group associations (when available), proper netmasks, and other information won't be processed the right way. In all cases, Object Filler always tries to figure out if the current file is of the right type. If the program detects that it's not, then will tell about this suspicious status, but this mistake detection mechanism is not 100% reliable, so always try to use the proper option.;-)

Why is the different the number of imported objects reported by Object Filler than the exported ones reported by Object Dumper? I compared them while running over the CSV file that Object Dumper just exported from my SmartCenter…
Most of the times, duplicates. Object Dumper doesn't apply too much verification while exporting objects, but Object Filler does while importing them. So, if you have the same IP address under different names, or the same port number under different names for example, Object Filler will process just the first one found and will complain and report the others as duplicates/invalids. I know these kind of duplicates are something totally permissible by SmartCenter and valid from the operations point of view, however, I just wanted to make sure that people knows (once more) they have duplicates, while importing. ;-)

How do I create a CSV file with Microsoft Excel to import it later with Object Filler?
Just create a new spreadsheet, and follow the column order described previously on this document.

Instead of saving it as a usual spreadsheet, select "Save As" and then choose the CSV Format (usually represented as "Comma Separated Values", "CSV File" or something like it). If you do this, please just make sure that the resulting file doesn't have quote signs (") also as field separators. If it has quote signs, then just remove them. If you don't remove such quote signs, it will result in problematic behavior of the tool.

Are you going to support *X* type of object?
May be. Depends on feedback... – If the tools currently don't support an object type you need, please let me know...

Why the tools don't simply support all the known type of objects for once?
For 2 basic reasons: I want to keep the code as simple as possible. Keeping support for all (even rarely used) known objects, adds complexity to the code (making it harder to maintain) and to the usability of the tools. That's why I rely on your feedback to do something extra. In the other hand, some objects may change (definition, naming convention, properties) while the most used ones are less likely to change. Having fewer changes in the code leads (again) to more stable tools, and more usability on them.

Why Object Filler doesn't support users?
Because there's already a way to do so, it's officially supported and it's well documented: Use "fw dbimport" and "fw dbexport" for that. The SmartCenter and the Command Line documentation have good information on this.

3. Object Dumper

So, What's the main purpose of Object Dumper? Doing backups or migrations?
No. Object Dumper was created more to assist *a bit* on documentation stages, but mainly to make day-to-day operations easier in conjunction with Object Filler, especially on bulk imports, modifications or transports. When Object Filler needs some information from the current configuration to do a job, Object Dumper is supposed to provide this. Definitely I did not have in mind backups nor complete migrations purposes, even though I've got several reports of people using it for real world migrations on relatively simple (even though some of them real big, with hundreds of thousands of objects) installations.

Why Object Dumper is not good for doing backups, policy merges, upgrades or migrations?
First, because it's not supported and you want to have Check Point support backing you if something goes wrong. But also, because Object Dumper won't give you important information you would need in a restore case. For example: Object Dumper won't export important object properties such as certificates or particular VPN settings. For backups and migrations the cpmerge, upgrade_export and upgrade_import tools available at Check Point's web site, are by far much better: more powerful, more easy to use, more focused precisely on that, and especially they are officially supported. You can find those tools and their respective documentation here:
http://www.checkpoint.com/techsupport/downloadsng/utilities.html

Does it work with 4.1 objects.C files?
Yes, it does work. I've done some testing with VPN-1/FireWall-1 4.1 objects.C files (located under $FWDIR/conf/objects.C) and it works recognizing hosts, networks and some services, but it has not been fully tested. This is only for objects of types hosts and networks , TCP and UDP services. Rulebases file from 4.1 Check Point products has not been tested at all.

Why Object Dumper doesn't support users?
Because there's already a way to do so without it, it's supported and it's well documented: Use "fw dbexport" and "fw dbimport" for that. Please refer to the Check Point SmartCenter and Command Line documentation for more information on such commands.

4. Common problems

Why I can't see CSV files on columns when I open them with Microsoft Excel?
Try this: Go to "Data" Menu, choose "Get External Data" or "Import External Data", then "Import Text File" or "Import Data". Please select the file you're trying to import and press "Import or

"Open". If you're on Microsoft Office Excel 2000, then choose "Delimited", then press "Next". Now choose "comma" and then press "Finish". If any additional windows appear, just press "OK". That should do the trick.

I'm getting the error *"'@'network_objects' - Token contain illegal character - Invalid Object Name* while trying to import the dbedit file produced by Object Filler. I check the file and it seems to be ok. What is going on?
Almost for sure you're transferring the file to another machine in a different platform/operating system. Please make sure that while you're doing so, you're transferring the file as ASCII TEXT. If you're doing the transfer via FTP, remember that some clients/servers behave with Binary (bin) transfer mode as default. Change the transfer mode to TEXT (ASCII) before transferring the file. Using the TEXT (ASCII) mode to transfer the file will fix the problem, if this is the cause. If it is not your case, please let me know.

I'm getting an error while importing the dbedit file, that says *Error... syntax error in line NNN Aborting.* - I look in the file for such NNN line number, and it's a blank line. What can be wrong?
The blank line itself is wrong. The dbedit utility will always complain if a blank line is found in the file used to specify commands to be executed. Usually this kind of thing happens when you copy-paste the contents of the dbedit commands file that results from the Object Filler execution. Normally Object Filler doesn't append this blank line. If you add it by accident while copying-pasting, you may safely ignore this message.

What does error *network_objects::XXXXX Object XXXXX already exists* or *Object XXXXX already exists* means? – I get it eventually while I'm importing the objects
This error means that dbedit got the instruction to create an object that already exists. This error is very frequent when you export the current configuration via Object Dumper, and then try to import back modifications in the objects via Object Filler, but forget to change the object type to a *mod* object (i.e. use *modhost* instead of *host*, *modnet* instead of *net*, etc.)

While I'm importing a big dbedit file I get several errors in a row. The messages say something like *A disk error occurred during a write operation*, *Failed To Send Audit Log*, *Failed To Send Audit Log for network_objects:: XXXXX* or *network_objects::XXXXX Object XXXXX already exists* – I also notice that not all the objects/rules that are supposed to be processed, were taken correctly. I have a large (hundreds, thousands) number of lines in the dbedit commands file being used. What can be happening?
You are processing way to many objects at the same time. You have two alternatives: split the processing (the dbedit commands file being processed) into smaller pieces, or open manually a dbedit session, and then copy-paste a reasonable amount of lines (200 or 250 are okay) at the same time. Wait until that is processed, and continue with the following ones…

Appendix B. Valid colors for objects in Object Filler

The following is the list of valid colors to be specified in the command line for Object Filler, as well as in the CSV file format.

aquamarine1	gold3
black	gray
blue	gray83
blue1	gray90
brown	green
burlywood4	lemonchiffon
coral	lightseagreen
cyan	lightskyblue4
darkorange3	magenta
darkseagreen3	medium
deepskyblue1	orange
dodgerblue3	pink
firebrick	red
Foreground	sienna
gold	yellow

When exporting objects with Object Dumper, due some limitations on the program, some times the color may be exported wrongly. On these cases, the object's color will be reset to black.

Appendix C. Default objects recognized by Object Dumper and Object Filler.

The following Object names are recognized as default (predefined) objects by Object Filler and Object Dumper. Names are NOT case-sensitive for the tools, which means that there's no difference between telnet, Telnet or TELNET.

In Object Filler, these objects are recognized as previously processed by –nopv switch while importing rules.

In Object Dumper, these objects will not be reported in a processing, unless the –d (default) switch is used.

IP Address Ranges

DAG_range

Dynamic Objects

LocalMachine
InternalNet
DMZNet
AuxiliaryNet

UDP Services

archie
biff
Blubster
bootp
Citrix_ICA_Browsing
CP_SecureAgent-udp
CU-SeeMe
daytime-udp
dhcp-rep-localmodule
dhcp-req-localmodule
Direct_Connect_UDP
discard-udp
domain-udp
E2ECP
echo-udp
eDonkey_4665
FreeTel-outgoing-server
FW1_load_agent
FW1_scv_keep_alive
FW1_snmp
GNUtella_rtr_UDP
GNUtella_UDP
GTPv0
GTPv1-C
GTPv1-U
H323_ras
H323_ras_only

TCP Services

AOL
AP-Defender
AT-Defender
Back_Door_Setup
Backage
BackDoor-G
Citrix_ICA
Connect-Back_Backdoor
ConnectedOnLine
CP_Exnet_PK
CP_Exnet_resolve
CP_redundant
CP_reporting
CP_rtm
CPD
CPD_amon
CPMI
CrackDown
CreativePartnerClnt
CreativePartnerSrvr
DaCryptic
DameWare
daytime-tcp
DerSphere
DerSphere_II
Direct_Connect_TCP
discard-tcp
domain-tcp
echo-tcp
eDonkey_4661
eDonkey_4662
Entrust-Admin
Entrust-KeyMgmt
exec
finger
Freak2k

HackaTack_31789
HackaTack_31791
Hotline_tracker
ICQ_locator
IKE
interphone
ISAKMP
Kerberos_v5_UDP
kerberos-udp
L2TP
MetaIP-UAT
microsoft-ds-udp
MSN_Messenger_1863_UDP
MSN_Messenger_5190
MSN_Messenger_Voice
MSSQL_resolver
MS-SQL-Monitor_UDP
MS-SQL-Server_UDP
name
nbdatagram
nbname
NEW-RADIUS
nfsd
NoBackO
ntp-udp
OnTime
pcANYWHERE-stat
RADIUS
RainWall_Daemon
RainWall_Status
RainWall_Stop
RDP
RexxRave
rip
RIPng
securid-udp
sip
sip_any
snmp
snmp-read
snmp-trap
Streamworks
SWTP_Gateway
SWTP_SMS
syslog
TACACS
tftp
time-udp
tunnel_test
udp-high-ports
vosaic-data

ftp
ftp-bidir
ftp-pasv
ftp-port
FW1
FW1_amon
FW1_clntauth_http
FW1_clntauth_telnet
FW1_CPRID
FW1_cvp
FW1_ela
FW1_ica_mgmt_tools
FW1_ica_pull
FW1_ica_push
FW1_ica_services
FW1_key
FW1_lea
FW1_log
FW1_mgmt
FW1_netso
FW1_omi
FW1_omi-sic
FW1_pslogon
FW1_pslogon_NG
FW1_sam
FW1_sds_logon
FW1_sds_logon_NG
FW1_snauth
FW1_topo
FW1_uaa
FW1_ufp
GateCrasher
GNUtella_rtr_TCP
GNUtella_TCP
gopher
GoToMyPC
H323
H323_any
HackaTack_31785
HackaTack_31787
HackaTack_31788
HackaTack_31790
HackaTack_31792
Hotline_client
http
https
ICKiller
ident
IKE_tcp
imap
iMesh

vosaic-ctrl
VPN1_IPSEC_encapsulation
WebTheater
who
WinMX
Yahoo_Messenger_Voice_Chat_UDP

Other Services
AH
backweb
egp
ESP
FreeTel-incoming
FreeTel-outgoing-client
ftp_mapped
FW1_Encapsulation
ggp
gre
gtp_path_mgmt
gtp_reverse
gtp_v0_path_mgmt
gtp_v1_path_mgmt
http_mapped
icmp-proto
igmp
igrp
ospf
rip-response
Sitara
SKIP
smtp_mapped
traceroute
tunnel_test_mapped
vrrp
X11-verify

ICMP Services
dest-unreach
echo-reply
echo-request
ICMP_frag_needed
info-reply
info-req
mask-reply
mask-request
param-prblm
redirect
source-quench
time-exceeded
timestamp
timestamp-reply

InCommand
IPSO_Clustering_Mgmt_Protocol
irc1
irc2
IS411-srvr
Jade
Kaos
KaZaA
Kazaa
Kerberos_v5_TCP
kerberos-tcp
Kuang2
ldap
ldap-ssl
login
lotus
lpdw0rm
Madster
microsoft-ds
Mneah
MSN_Messenger_File_Transfer
MSNP
MS-SQL-Monitor
MS-SQL-Server
Multidropper
Napster_Client_6600-6699
Napster_directory_4444
Napster_directory_5555
Napster_directory_6666
Napster_directory_7777
Napster_directory_8888_primary
Napster_redirector
nbsession
NCP
netshow
netstat
nfsd-tcp
nntp
ntp-tcp
OAS-NameServer
OAS-ORB
OpenWindows
Orbix-1570
Orbix-1571
pcANYWHERE-data
pcTELECOMMUTE-FileSync
pop-2
pop-3
Port_6667_trojans
pptp-tcp
RainWall_Command

RPC Services

cachefsd
cmsd
mountd
nfsprog
nisplus
nlockmgr
pcnfsd
rstat
rwall
sadmind
snmpXdmid
statd
ttdbserverd
ypbind
yppasswd
ypserv
ypupdated
ypxfrd

DCE-RPC Services

ALL_DCE_RPC
DCOM-RemoteActivation
HP-OpCctla
HP-OpCctla-bulk
HP-OpCctla-cfgpush
HP-OpCdistm
HP-OpCmsgrd-coa
HP-OpCmsgrd-m2m
HP-OpCmsgrd-std
MSExchangeADL
MSExchangeDirRef
MSExchangeDirRep
MSExchangeDSNSPI
MSExchangeDSRep
MSExchangeDSXDS
MSExchangeIS
MSExchangeMTA
MSExchangeStoreAdm
MSExchangeSysAtt
MSExchangeSysAttPriv
RAT
Real-Audio
RealSecure
Remote_Storm
rtsp
securidprop
Shadyshell
shell
smtp
SocketsdesTroie
sqlnet1
sqlnet2-1521
sqlnet2-1525
sqlnet2-1526
ssh
ssh_version_2
ssl_v3
StoneBeat-Control
StoneBeat-Daemon
SubSeven
T.120
TACACSplus
tcp-high-ports
telnet
Terrortrojan
TheFlu
time-tcp
TransScout
Trinoo
UltorsTrojan
uucp
wais
winframe
WinHole
X11
Xanadu
Yahoo_Messenger_messages
Yahoo_Messenger_Voice_Chat_TCP
Yahoo_Messenger_Webcams

Appendix D. Features Roadmap.

The following is a list of things that I think I'll include in the tools at some point of the time. They are not given in any particular order and there's no estimate on when they will be done. You may always send your suggestions and help prioritize this list by sending feedback.

Object Filler

- Support for security rules with User Groups as source, all the possible values in the track field and Resources within Services in CSV files
- Support for NAT rules in CSV Files and possibly Cisco PIX/Cisco Routers
- Support for NetScreen rules
- Support for SideWinder services
- Support for more file types from other firewall brands (?) – need sample files
- Support for a timestamp with created objects/rules
- Solaris native support via CLI
- Have a Windows and GNU/Linux native GUI
- Support different rulebases (names) from CSV files.
- Support colors with network object and service groups
- Support routes listing (netstat –nr) as a source file type for Object Filler
- Create a debug mode

Object Dumper

- Support for rules with User Groups as source, All the possible values in the track field and Resources within Services
- Support for NAT rules
- Support for a timestamp with dumper objects/rules
- Support for dumping rules only (without having to dump objects)
- Solaris native support via CLI
- Have a Windows and GNU/Linux native GUI
- Create a debug mode

Documentation

- Translate documentation to other Spanish and other languages. If there's people that volunteers for translating to other languages (or even Spanish ;-), such help would be always welcomed and acknowledged.
- Add images and screenshots, so the concepts may be clearer.