SECURITY
code

Secret Net Studio

Additional Tools

Mailing address:	**P.O. Box 66, Moscow, Russian Federation, 115127**
Phone:	**+7 495 982-30-20**
Email:	**info@securitycode.ru**
Web:	**https://www.securitycode.ru/**

Table of contents

Introduction

This manual is designed for Secret Net Studio administrators. It contains information that administrators need to use additional tools and configuration files (hereinafter additional tools) necessary to work with Secret Net Studio.

Website. Information about SECURITY CODE LLC products can be found on https://www.securitycode.ru.

Technical support. You can contact technical support by phone: +7- 800- 505- 30- 20 or by email support@securitycode.ru.

Training. You can learn more about hardware and software products of SECURITY CODE LLC in authorized education centers. The list of the centers and information about learning environment can be found on https://www.securitycode.ru/company/education/training-courses/. You can contact a company representative for more information about trainings by email education@securitycode.ru.

General information

If the standard management tools do not meet your needs or additional service operations are required, the additional tools allow you to configure and manage Secret Net Studio. The document contains a description and examples of the following additional tools usage:

- tools for working with the storage of centralized management objects;
- tools for working with the database management system (DBMS);
- tools for obtaining and saving data;
- tools for Integrity Check (IC) and Application Execution Control (AEC) mechanisms;
- tools for Mandatory Access Control and Discretionary Access Control subsystems;
- tools for the Device Control subsystem;
- other additional tools.

Tools for working with the CM object storage

SnDSTool.exe

SnDSTool.exe is used to perform the following operations with the storage of centralized management objects:

- clear the storage from the information about unused security tokens that remain, for example, after deleting a domain user with assigned security tokens on a computer without the Secret Net Studio client installed;
- obtain information about security domains;
- enable the **Trust Windows password authentication on the next logon** setting for the given user in the enhanced password authentication mode. When the network protection subsystem is installed, the necessary parameters are additionally set for synchronizing the user password with the authentication server data.

This tool is located in the Secret Net Studio distribution kit in the \Tools\SecurityCode\SnDSTool\Win32 or \Tools\SecurityCode\SnDSTool\x64 folders (depending on the OS version).

The tool runs in the command line on behalf of the current user. The command has the following input format:

```
SnDSTool.exe [-lds <server> <domainDN> [<port>]] [-ssl]
 -duei|-pds|-rpwd –u <domain\user> –a <administrator> –p <password>
```

Command description is provided in the table below.

Commands	Description
-?	Display information about using the tool
-lds <server> <domainDN> [<port>]	Connect to the specified LDS server. **<server>** — the name of LDS server. **<domainDN>** — the domain name of the main security domain in LDS. If these parameters are not specified, the tool will attempt to read them from the registry. **<port>** — the port number for connecting to the LDS server. If the standard port number 50002 is used by default or SSL protocol with the port number 50003 is used, the port is not specified
-ssl	Use SSL for LDAP connection
-duei	Delete the records about unused security tokens in the current security domain
-pds	Display information about all security domains
-rpwd -u<domain\user> -a <administrator> -p <password>	Enable the **Trust Windows password authentication on the next logon** setting for the given user by specifying: • **u <domain\user>** — the name of the domain user for whom you activate the setting; • **a <administrator>** — the LDS administrator name; • **p <password>** — the password of the LDS administrator

Command examples:

```
SnDSTool.exe –duei
```

Delete records about unused security tokens in the current security domain.

```
SnDSTool.exe –lds –duei
```

Delete the records about unused security tokens in the security domain whose connection settings are stored in the system registry.

```
SnDSTool.exe –lds LdsSrv –pds
```

Display the list of all security domains in the forest where the Security Server named LdsSrv is hosted.

```
SnDSTool.exe –rpwd –u Domain\Jones –a Administrator
–p Password
```

Enable the **Trust Windows authentication at the next logon** setting for domain user Jones. When the Network Protection subsystem is installed, the required settings for synchronizing the user password with the authentication server data are additionally set.

Tools for working with DBMS

Files for clearing the Security Server DB

Files for clearing the Security Server DB hosted on the MS SQL DBMS server (SQL server) include **clear.cmd**, **rebuild.cmd** command files and additional files. If the Security Server DB becomes full, the DB clearing procedure may be required to restore the operation of the SQL server.

This set of files is located in the Secret Net Studio distribution kit in the \Tools\SecurityCode\ClearMSSQL\ folder.

We recommend regularly archiving logs in the Security Server DB and performing other necessary actions to maintain an acceptable volume of this DB. The DB clearing procedure using the given files should be performed only if a DB overflow occurs and the Security Server cannot continue to function. As a result, all information stored in the DB will be lost, including the logs records for the centralized storage.

We also recommend periodically running the procedure of index rebuilding on the SQL server using the **rebuild.cmd** file. As a result of long-term operation and frequent archiving of the DB, server performance decreases
due to data fragmentation. The procedure of index rebuilding does not require stopping the operation of the server. However, it is recommended to run the command at the moments of the minimal load for optimal performance.

Attention! You need local administrator rights on the Security Server computer and DB administrator credentials on the SQL server to perform the DB clearing procedure.

To clear the Security Server DB:

1. On the computer with the Security Server, stop the Internet Information Services (IIS) and Secret Net Studio Security Server (server service).
2. On the SQL server, create a folder on the local disk. Copy the contents of the \Tools\SecurityCode\ClearMSSQL\ folder from the Secret Net Studio distribution kit to the created folder.
3. Open the copied files with the *.cmd extension for editing. Specify the DB administrator password set during the installation of the SQL server. Replace the default schema name **SN7_SERVER_SCHEMA** with the schema name used to install the Security Server. Enter the password instead of the **manager** substring.
4. In the **clear.sql** and **rebuild_index.sql** files, replace the default schema name **SN7_ SERVER_SCHEMA** with the schema name used when installing the Security Server. In the **clear.sql** file, the schema name must be written in square brackets.
5. Run the edited **clear.cmd** file. After successfully processing this file, run the **rebuild.cmd**.
6. Restart the Security Server.

Tools for collecting and saving information

SnDiagReport

SnDiagReport is used to collect diagnostic information that developers need to study problem situations.

This tool is located in the Secret Net Studio distribution kit in the \Tools\SecurityCode\SnDSTool\Win32 or \Tools\SecurityCode\SnDSTool\x64 folders (depending on the OS version). The bitness of the tool must correspond to the bitness of the installed product.

Attention! The tool requires local administrator rights to run the -i, -t, -d commands. Local user rights are required to run -h, -m, -e commands.

The tool runs in the command line on behalf of the current user. The command has the following input format:

```
SnDiagReport.exe [-<command> [parameters]] [-e]
```

Descriptions of main commands are provided in the table below.

Commands	Parameters	Description
-h	**-**	Display information about using the tool
-i	**-**	Collect all the files and data necessary for Secret Net Studio diagnostics in the **SnDiagInfo.zip** archive. The archive file is created in the user's folder **%temp%\SnDiagInfo<creation_date_and_time>**. This is the default mode of tool operation
-t	**on [<folder path>]**	Enable Secret Net Studio tracing with default parameters. You need to restart the computer to enable it. The **<folder path>** parameter is the location of the trace log records. Default folder: %ProgramData%\Security Code\Secret Net Studio\Logs
	off	Disable Secret Net Studio tracing. You need to restart the computer to disable it
-d	**-**	Setup for collecting system and user debugging information about an error that occurred during the program operation
-m	**-**	Display the Secret Net Studio tracing

Descriptions of additional commands are provided in the table below.

Additional commands	Description
-e	Exit the tool after executing the command. It can be combined with any command described in the table above

If the tool is launched without specifying commands, it switches to the default operation mode.

Examples of the commands:

```
SnDiagReport.exe –i –e
```

All the files and data necessary for Secret Net Studio diagnostics are collected and placed in the **SnDiagInfo.zip** file. The file is in the user folder **%temp%\SnDiagInfo<creation_date_and_time>**. The tool closes after the command is executed.

```
SnDiagReport.exe –t on C:\Datalogs
```

Secret Net Studio tracing is enabled with the default parameters. The trace log records are saved to C:\Datalogs. Tracing will be enabled after computer restart.

GetEventLog.exe tool

GetEventLog.exe is used to create Secret Net Studio log archives and shadow copies. It is also possible to clear the log and storage.

This tool is located in the Secret Net Studio distribution kit in the \Tools\SecurityCode\GetEventLog\Win32 or \\Tools\SecurityCode\GetEventLog\x64 folders (depending on the OS version).

Attention! A user with the log viewing privilege can create an archive of the Secret Net Studio log. A user with the log management privilege can clear the log after creating an archive.

The tool runs in the command line on behalf of the administrator. The command has the following input format:

```
GetEventLog.exe -n <file name> [-c|-s]
```

Command descriptions are provided in the table below.

Commands	Description
-?	Display information on using the tool
-n <file name>	Create an archive of the Secret Net Studio log in a file with the .evt or .evtx extension. The path to the file is specified in the file name. This is a mandatory parameter
-c	Clear the Secret Net Studio log and shadow copy storage after creating a log archive and a copy of the shadow copy storage. A shadow copy is created in the same folder as the Secret Net Studio log archive. This is an optional parameter
-s	Create a shadow copy. The Secret Net Studio log and shadow copy storage are not cleared. This is an optional parameter

Command examples:

```
GetEventLog.exe –n c:\EvtLog\SnEventLog.evtx –c
```

Create an archive of the Secret Net Studio log in a **SnEventLog.evtx** file and a shadow copy in the same folder. Clear the log and the shadow copy storage after the archive and the copy are created.

```
GetEventLog.exe –n c:\EvtLog\SnEventLog.evtx –s
```

Create an archive of the Secret Net Studio log in the **SnEventLog.evtx** file and a shadow copy. Do not clear the log and the shadow copy storage.

SnConfExport

SnConfExport is used to export the Client configuration with information about the status and policies of the subsystems, licenses used and TrustAccess settings. It is used on a computer with the Client installed.

This tool is located in the Secret Net Studio distribution kit in the \Tools\SecurityCode\SnConfExport\Win32 or \Tools\SecurityCode\SnConfExport\x64 folders (depending on the OS version). The bitness of the tool must correspond to the bitness of the installed product.

The tool runs in the command line on behalf of the current user. The command has the following input format:

```
SnConfExport.exe[<file path>]
```

The **<file path>** parameter is the path to the file where the configuration is saved. By default, the file is **SnConfExport.xml** in the tool startup folder.

Example of the exported configuration

The configuration consists of four parts.

1. Information about the subsystems state. The path in the configuration — OmsConfiguration/OmsObjects/OmsObject/SubsystemInfo.

   Depending on the subsystem, the information can be presented in two formats:

   - The information is contained in the **Subsystem** elements. The format is used for subsystems from the table below.

Subsystem	Name in the configuration
Virtual subsystem of the SNS core	SnCore
Data Wipe	SnEraser

Subsystem	Name in the configuration
Mandatory Access Control	SnMandat
Print Control	SnPrint
Device Control	SnDacs
Trusted Environment	SnExeQuota
Disk Protection	SnDiskProtection
Discretionary Access Control	SnFDC
Encryption of traffic from AD LDS	SnEncTraffic

The state of the subsystem is described by the parameters of the **SecuritySubsystem** element. If the subsystem is enabled, then an element with additional information about the state of the subsystem can be added to the **SecuritySubsystem** element. The name of such an element is the same as the name of the subsystem.

Parameter	Description
name	The name of the subsystem
state	The operation mode of the subsystem. It can take the following values: • On – enabled; • Off – disabled; • NotInstalled – not installed; • NotActivated – the system is present on the client, but it is not enabled locally; • Undefined – unknown (for example, errors occur while identification of the operation mode or information is not requested)
flags	Can be empty or take the following value: • NotManaged – the subsystem cannot be managed now
rebootRequired	Indicates that the computer needs to be restarted for the subsystem to work. It can take the following values: • true – a restart is required; • false – no restart is required

Example:

```
<SecuritySubsystem name="SnEraser" state="On" flags="" rebootRequired="false">
        <SnEraser eraseLocal="0" eraseRemovable="0" eraseMemory="0" eraseDemand="1" eraseDisk="1"/>
</SecuritySubsystem>
```

- The information is contained in the **Component** elements. The format is used for subsystems from the table below.

Subsystem	Name in the configuration
Firewall (for Windows)	FW
Network Authentication	NETAUTH
Intrusion detection and prevention (Network)	NIPS
Intrusion detection and prevention (host)	HIPS
Antivirus	AV
Software Passport	SOFTPSPT
Full Disk Encryption	FDE

Information about the state of the subsystem is in the child elements of the **Component** element.

Element	Description
Name	The name of the subsystem

Element	Description
Version	The version of the subsystem in the following format: <Major version>.<Minor version>.<build number>.<additional build number>.<flags>
Operation	The operation of applying information. It can take the following values: • new – complete information, it is necessary to completely replace the information from the recipient; • update- specific changes, it is necessary to update the sent element
ComponentState	The required state of the subsystem operation. It can take the following values: • On – enabled; • Off – disabled; • NotInstalled – not installed; • NotActivated – the system is present on the client, but it is not enabled locally; • Undefined – unknown (for example, errors occur while identification of the operation mode or information is not requested)
CurrentState	The current state of the security subsystem. It can take the following values: • undefined – the status cannot be defined; • stopped – stopped; • start_pending – starts; • stop_pending – stops; • running – started
RebootRequired	Indicates that the computer must be restarted for the subsystem to work. It can take the following values: • true – a restart is required; • false – no restart is required
Data	Settings and the state of the security subsystem in Base64 format. Additional information is in the elements **Data**/**States**/**State**: • the **Name** element contains the name of the section; • the **Value** element contains data about the subsystem in the form of xml encoded in Base64

Example:

```
<Component>
    <Name xmlns:dt="urn:schemas-microsoft-com:datatypes" dt:dt="string">FW</Name>
    <Version xmlns:dt="urn:schemas- microsoft- com:datatypes" dt:dt="string">8.7.2727.0.0</Version>
    <Operation xmlns:dt="urn:schemas- microsoft- com:datatypes" dt:dt="string">new</Operation>
    <ComponentState xmlns:dt="urn:schemas- microsoft- com:datatypes" dt:dt="string">On</ComponentState>
    <CurrentState xmlns:dt="urn:schemas- microsoft- com:datatypes" dt:dt="string">running</CurrentState>
    <RebootRequired xmlns:dt="urn:schemas- microsoft- com:datatypes" dt:dt="string">false</RebootRequired>
    <Data/>
</Component>
```

2. Subsystem policies. The path in the configuration is OmsConfiguration/OmsObjects/OmsObject/Policies.

   They are arranged according to the **Policies** elements by sections. The **type** attribute is responsible for the section name.

   Example:

```
<Policies type="Basic">
<Policies type="Dacs">
<Policies type="AV">
<Policies type="NIPS">
<Policies type="UPD">
<Policies type="SOFTPSPT">
```

3. Information about licenses. The path in the configuration is OmsConfiguration/SecurityDomains/SecurityDomain/Licenses.

   The information is in the child elements of the **License** element.

Element	Description
Data	The license text encoded in Base64
State	The state of the license
State/Code	The license error code. It can take the following values: • 0 — valid license; • not 0 — license with an error
State/Description	Description of the error. For a valid license, it contains **OK**
State/DaysToExpire	The number of days before the license expires. Examples values: • 0 – the license expires today; • #NaN# – the license never expires. If the **Code** element contains an error code - E_LicenseError_Overdue, the **DaysToExpire** element is not considered

   Example:

```
<License licId="330003">
        <Data dt:dt="string">XXXXX</Data>
        <State>
        <Code dt:dt="int">0</Code>
        <Description dt:dt="string">OK</Description>
        <DaysToExpire dt:dt="int">29</DaysToExpire>
        </State>
</License>
```

4. TrustAccess settings. The path in the configuration is OmsConfiguration/TrustAccess.

Tools for Integrity Check and Application Execution Control subsystems

SnIcheckCmdTool.exe

SnIcheckCmdTool.exe is designed to work with the Local IC-AEC database in which the data model is stored, and allows to do the following:

- start full synchronization of changes made via the central IC-AEC DB;
- prepare resources for AEC;
- view information about default group objects;
- recalculate resource benchmarks;
- update benchmark storages containing fixed benchmarks of Secret Net Studio resources for the **Contents** control method and the CRC32 algorithm (used during IC of Secret Net Studio resources and can be replaced only when authorized software updates of the protection system are installed).

The tool is located in the Client installation directory — C:\Program Files\Secret Net Studio\Client by default.

Attention! You need the rights of a local administrator to work with the utility.

The utility performs actions in the command line mode on behalf of the current user. The command line has the following format:

```
SnIcheckCmdTool.exe /<command> [<attribute> <object name>]
```

Descriptions of the commands are presented in the following table.

Commands	Description
-?	Display information about using the utility
/fullsync	Start the full synchronization of changes in the IC-AEC CDB
/fullsynccentral	Start the full synchronization of changes in the IC-AEC CDB using the function of sending notifications about changes for this computer (the computer must be present as a separate subject in the centralized data model)
/reloaduel	Launching the procedure of preparing resources for AEC. The procedure is performed for all users with opened work sessions on this computer at the given moment
/defgroup	Displaying the list of security identifiers (SID) of computers included in the **SecretNetICheckDefault** or **SecretNetICheckDefault64** default group (depending on the OS bitness)
/recalc	Recalculate benchmarks of the specified resource for the **Contents** control method and the CRC32 algorithm and saving them in the IC-AEC LDB. To run the command, you need to specify additional attributes presented in the table below
/etalon	Write new benchmark values of the specified resource in the local database for Secret Net Studio distribution kit benchmarks. The database contains fixed benchmarks of all Secret Net Studio resources for the **Contents** control method and the CRC32 algorithm. To run the command, you need to specify additional attributes presented in the table below
/etalonxml	Write new benchmark values of the specified resource in the xml file with Secret Net Studio distribution kit benchmarks. To run the command, you need to specify additional attributes presented in the table below
/defmodel	Import the default model to the IC Database
/transformxml	Convert the initial xml files with the Secret Net Studio distribution kit data into a single formatted end file
/rebuild	Reopen scripts for default tasks

To run the tool with /recalc, /etalon, /etalonxml parameters, you need to specify the additional attributes. The information about using the tool with the specified parameters (launch mode with the description of the applied attributes) is displayed on running the tool with no attributes parameter. The following attributes are provided:

Attribute	Description
f <object name>	Execute the command for the specified file

Attribute	Description
c <object name>	Execute the command for the specified directory
k <object name>	Execute the command for the specified registry key
v <object name>	Execute the command for the specified registry parameter
a <object name>	Execute the command for all files in the specified directory
r <object name>	Execute the command for all parameters in the specified registry key
w <object name>	Used during the checksums calculation for the 32-bit executable files meant for using in the 64-bit operating systems (is not used for the /recalc parameter)

Examples of commands:

```
SnIcheckCmdTool.exe /recalc f snicheckapi.dll
```

Recalculate the benchmark value for the snicheckapi.dll file.

```
SnIcheckCmdTool.exe /recalc a c:\
```

Recalculate the benchmark values for all files in the C: root directory and save them to the IC-AEC LDB. The recalculation is performed for the benchmarks calculated using the CRC32 algorithm.

```
SnIcheckCmdTool.exe /etalonxml c C:\admin
```

Write new benchmark values of the specified admin directory to the xml file with the Secret Net Studio distribution kit benchmarks.

Tools for mandatory and discretionary access control subsystems

The SnMCUtil.exe utility

SnMCUtil.exe is meant for creating the list of paths to the redirect directories in the flow control mode and managing the user rights. It allows to do the following:

- checking the specified paths with the automatic creation of additional directories for different confidentiality categories (in case of absence of such directories for the specified paths);
- adding new paths to the list (creating the redirect rules);
- deleting the paths from the list (deleting the redirect rules);
- managing access level privileges of a user.

This tool is located in the Secret Net Studio distribution kit in the \Tools\SecurityCode\SnMCUtil\ folder. Depending on the OS bitness — in the Win32 and x64 subfolders.

You can launch the utility in a user session or in the context of a system account (for example, using Windows Task Scheduler or group policies). In a user session, the main utility functions are available if the same requirements for the work with the configuration program for the mandatory access control subsystem are met:

- a user is included in the local administrators group;
- a user is assigned the highest level of access to confidential information;
- a user is granted the **Confidentiality category management** privilege;
- the mandatory access control mechanism is enabled;
- flow control is disabled.

Attention! When launching in the context of a system account, the necessary requirement is only the enabled mandatory access control mechanism.

The utility performs actions in the command line mode on behalf of the current user. The command line has the following format:

```
SnMCUtil.exe -<command> -<parameter> [<argument>]
```

Descriptions of the commands are presented in the following table.

Commands	Parameters	Description
-?	**-**	Display information about using the utility
-redir	**-list**	Display the existing redirect paths
	-add <path to the directory> [-apply <path>]	Add a new redirect path. Paths for redirecting are added one by one. A line can contain both the full path explicitly defining this directory and the template (part of the path) allowing to determine a subset of paths to directories. A subset of paths needs to start with a "\" character, and the path to the directory is specified without a "\" character in the end. If you do not need to copy the files from the initial directory to the redirect directories — add the "**" characters at the end of the path. If you do not need to copy the subdirectories from the initial directory to the redirect directories — add the "*" characters at the end of the path. The additional argument: **-apply** <path> — perform checking and processing for the new redirect path. The <path> parameter can be used to narrow the check area to a separate local disk (for example, C:\) or directory (C:\Users)
	-del <path to the directory>	Delete of the redirect path from the list. Paths for redirecting are deleted one by one
	-check <path to the directory>	Check the existing redirection paths and creating the missing directories or files

Commands	Parameters	Description
-user	**-get <username>**	Display of the current access level and privileges of a user. If the username is longer than 20 characters, you need to specify it in the following format: "long_user_name@domain"
	-set <username> [-level <access level>] [-privs <set of privileges>]	Changing the access level and privileges of a user. The additional argument: **-level <access level>** — the new access level of a user. The number of categories depends on the system settings. There are 3 confidentiality categories by default: • 0 – Non-confidential; • 1 – Confidential; • 2 – Strictly confidential. The additional argument: **-privs <set of privileges>** — the new privileges of a user. Can take the following values: • value not defined – cancel all existing privileges of a user; • ConfManage – confidentiality categories management; • ConfOutput – confidential information output; • ConfPrint – confidential information print

Examples of commands:

```
SnMCUtil.exe -redir -add "\appdata\local\roaming\microsoft product"
```

Add a new redirection rule for the **\appdata\local\roaming\microsoft product** path template without creating the redirection directories themselves.

```
SnMCUtil.exe -redir -add "\appdata\local\roaming\microsoft product" -apply
```

Add a redirection rule for the **\appdata\local\roaming\microsoft product** path template, then the search on all local disk directories matching the template is performed, and the required redirect directories are created.

```
SnMCUtil.exe -redir –check c:\
```

Search disk C: for already set redirection paths and create missing redirection directories.

```
SnMCUtil.exe -user –get Peters
```

View current access level and privileges of user Peters.

```
SnMCUtil.exe -user -set Peters -level 2 -privs ConfManage ConfOutput
```

Assign **Strictly confidential** access level, and grant **Confidentiality categories management** and **Confidential information output** privileges to user Peters.

The SnSessLevel.exe utility

SnSessLevel.exe is meant for displaying the current confidentiality level of a user session. A value of **-1** — means that flow control is disabled.

This tool is located in the Secret Net Studio distribution kit in the \Tools\SecurityCode\SnSessionLevel\ directory.

SnSessLevel.exe performs actions in the command line mode on behalf of the current user. The tool has no parameters.

SetSecAttrib.exe

SetSecAttrib.exe is meant for managing the mandatory and discretionary access parameters of directories and files.

The utility is located in the Client installation directory — C:\Program Files\Secret Net Studio\Client by default. You can launch it only from this directory.

Manage mandatory access settings

To change the mandatory access settings of a directory or file, a user must have the **Confidentiality category management** privilege. Without it a user can only raise the categories for files, but not higher than his/her own access level, user session confidentiality level and confidentiality category of the directory.

The tool performs actions in the command line mode on behalf of the current user. The command line has the following format:

```
SetSecAttrib.exe <Resource name> [-l <category>]
[-f <flag>] [-r <recursion type>]
```

Descriptions of the commands are presented in the following table.

Command	Description
-?	Display information about using the utility
<Resource name>	Specify the full path to the file or directory to which confidentiality category and inheritance flags are assigned
-l <category>	Assign a confidentiality category to a resource. The number of categories depends on the system settings. There are 3 confidentiality categories by default: • 0 – Non-confidential; • 1 – Confidential; • 2 – Strictly confidential
-f <flag>	Assign the inheritance flags. Only for directories. The <flag> parameter can take the following values: • no flags set – delete all inheritance flags; • IF – set the "inherit for files" flag; • IS – set the "inherit for directories" flag
-r <recursion type>	Execute commands for the child objects of the directory. The <recursion type> parameter can take the following values: • F – processing of the specified directory and the files in it; • S – processing of the specified directory and the subdirectories excluding the files contained in them; • SF – processing of the specified directory, the subdirectories and all files contained in them

Examples of commands:

```
SetSecAttrib.exe C:\folder\file.txt
```

Display current access settings of the **file.txt** file.

```
SetSecAttrib.exe C:\folder\file.txt –l 1
```

Assign the **Confidential** category to the **file.txt** file.

```
SetSecAttrib.exe C:\folder –f IF IS
```

The inheritance flags are set for the C:\folder directory. As a result, the confidentiality category of the directory will automatically be assigned to all subdirectories and files created in it.

```
SetSecAttrib.exe C:\folder –l 2 –f IS –r SF
```

Example of using recursion. The C:\folder directory, all its files, subdirectories and files in them are assigned the Strictly confidential category. Also an inheritance flag requiring the automatic assignment of the confidentiality category of the directory to all subdirectories created in it is set for this directory and all its subdirectories.

Manage discretionary access settings

To change the discretionary access parameters of resources, a user must have the **Access rights management** privilege or a permission to manage the access privileges for this resource.

The utility performs actions in the command line mode on behalf of the current user. The command line has the following format:

```
SetSecAttrib.exe <Resource name> [-<command> <parameter block 1>;<parameter block 2>;…<parameter block N>] … [-r <recursion type>]
```

Descriptions of the commands and parameters are presented in the following table.

Commands	Parameters	Description
-?	-	Getting information about using the utility
<Resource name>	-	Specifying the full path to the file or directory to which access and audit privileges are assigned

Commands	Parameters	Description
-s	**<user or group>:<rule type>(<access types>)**	Assigning new access privileges to a resource instead of the existing ones. At the same time, inheritance of access and audit privileges is disabled. The current audit privileges are fully saved. Access privileges are set by a block of 3 parameters. You can specify several such blocks separated by the ";" sign. The <user or group> parameter — account security identifier (SID) or the full name of a user or group. The <rule type> parameter — you can specify only one value: "+" for permission or "-" for prohibition. The <access types> parameter — a list of allowed or prohibited operations. A set of values: • R – reading; • W – writing; • X – execution; • D – deletion; • P – access privileges management
-sa	**<user or group>:<audit type>(<access types>)**	Assigning new audit rules for a resource instead of the existing ones. At the same time, inheritance of audit rules and access privileges is disabled. The current access privileges are fully saved. Audit rules are set by a block of 3 parameters. You can specify several such blocks separated the ";" sign. The <user or group> parameter — SID or the full name of a user or group. The <audit type> parameter — you can specify one or both values: "+" for success, "-" for denial (or "+-"). The <access types> parameter — a list of operations for audit. A set of values matches the set for the -**s** command
-g	**<user or group>:(<access types>)**	Adding permissions to the active access privileges. At the same time, inheritance of access privileges and audit rules is disabled. The current audit rules are fully saved. Permissions are set by a block of 2 parameters. You can specify several such blocks separated by the ";" sign. The <user or group> parameter — SID or the full name of a user or group. The <access types> parameter — a list of allowed operations. A set of values matches the set for the -**s** command
-d	**<user or group>:(<access types>)**	Adding prohibitions to the active access privileges. At the same time, inheritance of access privileges and audit rules is disabled. The current audit rules are fully saved. Prohibitions are set by a block of 2 parameters. You can specify several such blocks separated by the ";" sign. The <user or group> parameter — SID or the full name of a user or group. The <access types> parameter — a list of prohibited operations. A set of values matches the set for the -**s** command
-a	**<user or group>:<audit type>(<access types>)**	Adding audit rules to the active rules. At the same time, inheritance of audit rules and access privileges is disabled. The current access privileges are fully saved. Audit rules are set by a block of 3 parameters. You can specify several such blocks separated by the ";" sign. The <user or group> parameter — SID or the full name of a user or group. The <audit type> parameter — you can specify one or both values: "+" for success, "-" for denial (or "+-"). The <access types> parameter — a list of operations for audit. A set of values matches the set for the -**s** command
-c	-	Enabling the inheritance of access privileges and audit rules from a parent directory mode for a specified resource. The current access privileges and audit rules are deleted
-r	**<recursion type>**	Executing commands for child objects of the directory too. The <recursion type> parameter can take the following values: • F – processing of the specified directory and the files in it; • S – processing of the specified directory and its subdirectories excluding the files contained in them; • SF – processing of the specified directory, the subdirectories and all files contained in them

Examples of commands:

Setting new access privileges and audit rules:

```
SetSecAttrib.exe C:\folder\file.txt -s S-1-1-0:-(WD);
BUILTIN\Administrators:+(RWXDP)
```

The new access privileges are set for the file.txt file instead of the existing ones and the inheritance mode is disabled (if it was enabled). The user with SID S-1-1-0 is prohibited to perform "writing" and "deletion" operations. The Administrators group is allowed to perform all operations. Audit rules are not changed.

```
SetSecAttrib.exe C:\folder –sa DOMAIN\Ivanov:+-(RWX)
```

The new audit rules are set for the folder directory instead of the existing ones and the inheritance mode is disabled (if it was enabled). All successful and unsuccessful attempts to perform "reading", "writing" and "execution" operations will be registered for the domain user DOMAIN\Ivanov. Access privileges are not changed.

```
SetSecAttrib.exe C:\folder\file.txt -s S-1-1-0:+(RWXD) –sa S-1-1-0:-(RWXD)
```

An example of using commands for setting access privileges and audit rules in one command line.

Changing access privileges and audit rules:

```
SetSecAttrib.exe C:\folder –g DOMAIN\Ivanov:(P)
```

Permissions are added to the existing access privileges for the folder directory and the inheritance mode is disabled (if it was enabled). The domain user DOMAIN\Ivanov is allowed to manage access privileges for this directory now. Audit rules are not changed.

```
SetSecAttrib.exe C:\folder\file.txt -d S-1-1-0:(WD)
```

Prohibitions are added to the active access privileges for the file.txt file and the inheritance mode is disabled (if it was enabled). The user with SID S-1-1-0 is prohibited to perform "writing" and "deletion" operations. Audit rules are not changed.

```
SetSecAttrib.exe C:\folder -a DOMAIN\Ivanov:+-(X)
```

Audit rules are added to the active rules for the folder directory and the inheritance mode is disabled (if it was enabled). All successful and unsuccessful attempts of launch in the directory of executable files will be registered for the domain user DOMAIN\Ivanov. Access privileges are not changed.

Enabling inheritance:

```
SetSecAttrib.exe C:\folder –c
```

The inheritance of access privileges and audit rules from a parent directory mode is enabled for the folder directory. Current access privileges and audit rules are deleted.

Additional examples:

```
SetSecAttrib.exe C:\folder –g DOMAIN\Ivanov:(P) –r S
```

An example of using recursion. New permissions are added for the folder directory and all its subdirectories in addition to the active access privileges.

```
SetSecAttrib.exe C:\folder –l 2 –f IS –g S-1-1-0:(RWX) –r F
```

An example of using commands for managing the mandatory and discretionary access parameters in one command line.

Tools for the Device Control subsystem

SnHwUtil.exe

SnHwUtil.exe is meant for working with the list of devices of the computer and provides the following features:

- approving detected changes in the device configuration;
- checking changes in the device configuration;
- loading the current list of devices;
- searching and fixing invalid records in the device list;
- deleting explicitly specified control parameters and access privileges in the device list;
- deleting devices which are absent on the computer from the list;
- export of the device list to a file.

This utility is located on the Secret Net Studio installation CD in the \Tools\SecurityCode\SnHwUtil\ directory. Depending on the OS bitness — in the Win32 and x64 subdirectories.

Attention! You need the rights of a local administrator to access the device list.

The utility performs actions in the command line mode on behalf of the current user. The command line has the following format:

```
SnHwUtil.exe -<command>
```

Descriptions of the commands are presented in the following table.

Command	Description
-?	Getting information about using the utility
-c	Approve the equipment configuration
-q	Check the equipment configuration
-s	Update the device list
-v	Find and fix invalid records in the device list
-l	Show the list of devices with unique names
-r	Switch the control parameters of all devices to the **Not controlled** value
-h	Set the values of the Device Control parameters or the Mandatory Access Control parameters for the group or class of devices
-i [<File name>]	Import a device from the .sndev file(s)
-a <Group name or class name>	Enable the inheritance of parameters for the devices in a group or a class
-n [<File name>]	Show the record about the device from the .sndev file
-m	Move the device to a new class
-t [<File name>]	Set the values of device control parameters using the configuration file
-d [-g]	Delete the devices absent on the computer from the device list
-f [<File name>]	Export the local device policy to a file
-e [<File name>]	Export the local device database to a file
-p [<File name>]	Export the local printer database to a file

Other tools

SnetPol.exe

The SnetPol.exe utility is meant for export and import of the parameters of effective (resultant) policy on the computer. Export/import is performed using the template files for group policies which format matches Windows information files (*.inf).

This utility is located in the distribution kit in the \Tools\SecurityCode\SnetPol\ directory. Depending on the OS bitness — in the Win32 and x64 subdirectories.

Attention! You need the rights of a local administrator to access the policy parameters.

The utility performs actions in the command line mode on behalf of the current user. The command line has the following format:

```
SnetPol.exe -<command>
```

Descriptions of the commands are presented in the following table.

Command	Description
-h	Display information about using the utility
-i <file name>	Import policy settings from a template file
-e <file name>	Export of policy settings to a template file and to xml files of policy blocks
-x <security component> <file name>	Load policy settings for a specified security component from xml file into an effective security policy. The following abbreviations are used for security components: AV - antivirus, NIPS - intrusion detection and prevention, UDP - updating components, SOFTPSPT - software passport

Command example:

```
SnetPol.exe –x AV "c:\AV.xml"
```

Policy settings for Antivirus are loaded from the AV.xml file.

```
SnetPol.exe –i "c:\test.inf"
```

Policy settings are imported from the test.inf file.

SnFCUtil.exe

The SnFCUtil.exe utility is meant for configuring the control subsystem for access to files and directories.

This utility is located on the installation CD in the \Tools\SecurityCode\SnFCUtil\ directory. Depending on the OS bitness — in the Win32 and x64 subdirectories.

The utility contains one command and performs actions in the command line mode on behalf of the current user.

Command example:

```
SnFCUtil.exe -base -fix
```

The restoration of the mapping to logical volumes of the local resource databases is performed.

Sns.av_cli.exe

Sns.av_cli.exe is meant to control the Secret Net Studio antivirus.

The tool is located in the Client installation directory — C:\Program Files\Secret Net Studio\Client\Components\Antivirus Protection by default.

Attention! Sns.av_cli.exe is designed for technical support specialists. We DO NOT RECOMMEND to use the tool for routine Antivirus configuration.

To display detailed information about the program, open the command line and enter the following command:

```
sns.av_cli.exe
```

To display quarantined objects, enter the following command:

```
sns.av_cli.exe -c:list_quarantine_objects
```

This command will display the list of quarantined objects with their IDs.

The following quarantine management commands are available only to the administrator .

Delete files from quarantine:

```
sns.av_cli.exe -c:remove_file_from_quarantine
-quarantine_file_id:<file ID>
```

Example:

```
sns.av_cli.exe -c:remove_file_from_quarantine
-quarantine_file_id:1
```

Delete old files from quarantine:

```
sns.av_cli.exe -c:remove_files_from_quarantine_older_than
-days:<number of days>
```

Example:

```
sns.av_cli.exe -c:remove_files_from_quarantine_older_than
-days:2
```

Restore a file from quarantine:

```
sns.av_cli.exe -c:restore_file -p:"<file path>"
```

Example:

```
sns.av_cli.exe -c:restore_file -p:"c:\checkAV\test
heuristic\heur\!ITW#460.vxe.quarantine"
```

```
sns.av_cli.exe -c:restore_file -p:"\\computer\
open_share\!test for localize\!ITW#460.vxe.quarantine"
```

This tool allows the administrator to restore files from quarantine even if the computer is not connected to the network or the file cannot be restored via the Secret Net Studio Control Center.

Files quarantined from an external drive can be restored on any computer. To restore them, install Secret Net Studio Antivirus, run Sns.av_cli.exe, execute the restore a file from quarantine command and enter the path to the .quarantine file.

SnUserImport.exe

SnUserImport.exe is meant for import of user accounts from the Windows AD database to the security system database (hereinafter — the SN database).
This utility is included in the distribution kit and is located in the \Tools\SecurityCode\SnUserImport\ directory. Depending on the OS bitness — in the Win32 and x64 subdirectories.

Attention!
- You can use the utility only in the network mode of operation.
- You need the rights of a local and security domain administrator to work with the utility.
- You can import only accounts that exist in Windows AD.

The utility performs actions in the command line mode on behalf of the current user. The command line has the following format:

```
SnUserImport.exe -<command> <parameter> -o
```

Command descriptions are presented in the table below.

Command	Description
-u <user name>	Import a user to the SN database and display the results
-f <file name>	Import a user list from an .csv file to the SN database and display the results

Additional command descriptions are presented in the table below.

Additional command	Description
-o <file name>	Save import results to a .txt file

Command examples:

```
SnUserImport.exe -u Smith@domain.ru
```

Imports user Smith to the SN database.

```
SnUserImport.exe -f users.csv -o results.txt
```

Imports the user list from users.csv to the SN database and saves the results to results.txt.

Snsshell.exe

Snsshell.exe is designed to manage the security system using the command line. The administrator may use this tool in case of an unforeseen situation (for example, if the Control Center is unavailable), or to partially automate the configuration of the security system. The tool allows to do the following:

- switch the self-protection mechanism to emergency mode;
- enable and disable the integration of the leak detection module into the Print Control mechanism.

The tool is located in the Client installation directory — C:\Program Files\Secret Net Studio\Client by default.

The tool performs actions on behalf of the current user. The command line has the following format:

```
snsshell.exe <component_name> <command> [-parameter1]
[-parameter2:value]... [-parameterN:”compound value”]
```

Command descriptions are provided in the table below.

Component	Command	Possible parameters	Required privileges	Description
selfprot\ printctrl (not required)	help	None	None	Display help for the tool or for specific components
selfprot	deactivatesd	None	• Local administrator • PIN	switch self-protection to emergency mode
printctrl	ldm	-on\-off	• Local administrator • PIN (when administrative privilege control is enabled)	enable and disable the integration of the leak detection module into the Print Control mechanism

Command examples:

```
snsshell.exe selfprot deactivatesd
```

Switches self-protection to emergency mode

```
snsshell.exe printctrl ldm -off
```

Disables the integration of the leak detection module into the Print Control mechanism

CitrixConfig

CitrixConfig.cmd is meant for configuring the process of preparing a base image for the Citrix PvD system. This utility is located on the installation CD in the \Tools\SecurityCode\CitrixConfig\ directory.

Attention! When Secret Net Studio self-protection is enabled, this operation is unavailable. Before using CitrixConfig.cmd, switch self-protection to emergency mode or disable it.

Ngeninstall

Ngeninstall.cmd is meant for static compilation of the control program into the target platform code in order to increase the speed of its launch. Compilation is performed automatically when the control program is installed. However, the compiled files can be deleted when updating the control program. This will decrease the speed of the control program launch. In this case, we recommend recompiling the control program by calling the ngeninstall.cmd file.

The file is run as a local administrator. Ngeninstall.cmd is located in the distribution kit in the \Tools\SecurityCode\Ngeninstall\ directory.