SoftControl

SysWatch 6.1.398

Test Procedure

Dear user!

ARUDIT SECURITY, LLC thanks you for choosing SoftControl SysWatch. Specialists of the company do their best to make sure our software both meets the highest requirements in a field of information protection and is easy use. We hope you find SoftControl SysWatch helpful.

Postal address:

127106 Russia, Moscow

Novovladykinsky passage, house 8, building 3

ARUDIT SECURITY, LLC

Tel:

+7 499 201-55-12

Fax:

+7 499 201-55-12

E-mails:

Customer service: support@safensoft.com

Sales team: sales@safensoft.com

Website: safensoft.com

Contents

1. Introduction

SoftControl SysWatch is designed to protect Microsoft® Windows®-controlled computers from un-authorized data access.

This document describes how to test the installed software.

2. Preparing the test bench

You need two computers for the testing. They can be either physical or virtual machines, and should have USB ports. The computers should be able to connect to each other through the network. Install SoftControl Service Center on the first computer ('the server'; see 'SoftControl Service Center administrator's guide'). Install SoftControl SysWatch on the second computer ('the client computer'; see 'SoftControl ATM Client / Endpoint Client / SClient user's guide'). Collect the profile on the computer with SoftControl SysWatch. Connect SoftControl SysWatch to SoftControl Service Center.

Create two users on the client computer: *USER1* and *USER2*. Create the following directories on the *C:* drive of the computer: *No_rules*, *No_write_delete_all*, *No_write_program*, *Not_for_you*, *Not_now*. You will also need:

1) Two USB drives (*STOR1* and *STOR2*) with arbitrary contents.
2) Two file managers: Windows Explorer (*Explorer.exe*) and, for example, *FAR.exe* (http://www.farmanager.com/download.php?l=ru). Make sure that the second file manager is not an add-on for Windows Explorer (*FAR.exe* meets this condition).
3) Any two exe files, i.e. programs that do not require installation (*PROG1* and *PROG2*).
4) Seven installers, i.e. *exe* files (not *msi* files) that have the installer flags (see note below[5]):
   - an installer without digital signature (*INST1*);
   - the installers with valid digital signatures (*INST2*, *INST3*, *INST4*);
   - an outdated version of an installer (*INST5*) and an update package for it (*INST6*); both with valid digital signatures;
   - an outdated version of an installer of an application that can update itself (for example, a web browser) (*INST7*).

Note

SoftControl SysWatch work with the installers as follows.

1. When any executable file runs, SoftControl SysWatch checks whether its checksum is in the profile. If the checksum is in the profile, SoftControl SysWatch allows the file to run.
2. If the checksum is not in the profile, SoftControl SysWatch checks whether the file has any installer flags. An installer flag is a file name that contains *setup*, *install* or *update*, or the **Installer**

option in the file properties. If there is at least one of these flags, the file is considered an installer.

3. If the file is considered an installer, SoftControl SysWatch checks whether it has a valid digital signature. That means a signature that has either its certificate or any certificate from its certification path, in the Windows certificate store on this computer.
   a. If there is no signature or there is no signature's certificate in the certificate store, SoftControl SysWatch considers the installer an 'installer without valid signature' and blocks it.
   b. If the installer file has a valid signature and the whitelist of certificates in not enabled, SoftControl SysWatch allows the installer to run.
   c. If the whitelist of certificates is enabled, SoftControl SysWatch only grants the permission to run to the installers with a signature with the certificates that are in the whitelist and have the **Trust** option enabled.
4. When SoftControl SysWatch allows an installer to run, all the files that the installer has modified or created in the system are added to the profile. The installer file is added to the profile as well. It means that when the user tries to run the file next time, SoftControl SysWatch allows the file to run and does not check whether there are any installer flags in the file name or whether the file's signature certificate is in the whitelist.

3. Test cases

You should either perform all the tests twice (create the settings on the client computer and in SoftControl Admin Console), or perform some of the tests on the client computer and others through the organization unit settings in SoftControl Service Center. Some tests require settings that are only specified in the SoftControl SysWatch interface on the client computer. Such tests are marked as **Local settings only**.

3.1 Application and system integrity control

Table 1. Testing the application and system integrity control

Description	Expected result
Blocking unknown processes	
Run *PROG1*.	The application is blocked.
Running trusted processes	
Add *PROG1* to the system profile. Run *PROG1*.	The application runs successfully.
Blocking trusted processes with modified executable code	
1) Remove *PROG1*. Copy *PROG2* to the directory with *PROG1*, change the name of the *PROG2* executable module to the name of the removed executable module (*PROG1*) and run *PROG2*. 2) Remove the renamed executable module (*PROG2*). Copy *PROG1* to the original directory. Run *PROG1*.	After you modified the executable module's code (replaced *PROG1* with *PROG2*), *PROG1* is blocked. *PROG1* runs successfully after you restore the code to its original state. Note: in some cases, the application may still be blocked until you restart the *safensec.exe* system service.

3.2 Installer control

Table 2. Testing installer control

Description	Expected result
Blocking unsigned installers	
Run *INST1*.	The unsigned installer is blocked.
Blocking installers with invalid digital signature	
Open the *INST2* file properties and make sure that it has a valid digital signature. Open *INST2* with the help of *Notepad.exe*, insert a character and save the file. When you modify the file, its digital signature becomes	The installer with invalid digital signature is blocked.

Description	Expected result
invalid. Run *INST2*.	
Installing software with valid digital signature	
Run *INST3*.	The installer completes its task. The program it has installed is added to the profile.
Installing software with valid digital signature when the whitelist of certificates is enabled	
1) Tick off **Enable whitelist of certificates** in the client module's settings. The *INST4* installer's signature certificate and all the certificates from its certification path should not be in the whitelist. If they are in the whitelist, deselect **Trust** for them. 2) Run *INST4*. 3) Add the *INST4*'s signature certificate to the whitelist. 4) Run *INST4*.	Until you add the installer's signature certificate to the whitelist, the installer is blocked. After you add the signature certificate to the whitelist, the installer successfully installs the application. Note: please remember that after you add the certificate to the whitelist and run the installer, the installer is added to the system profile. If you remove the certificate from the whitelist, SoftControl SysWatch still allows the installer to run, because SoftControl SysWatch considers it a software from the system profile, not an installer.
Updating software with the help of an updater with valid digital signature	
1) Run *INST5* and install the application. Make sure the application is added to the profile and can run. 2) Run the update package of *INST6*. After the update, run the updated software.	The application installed by *INST5* runs successfully after the update.
Updating software that can update itself	
1) Run *INST7* and install the application. Make sure the application is added to the profile and can run. 2) Run the update from the application interface. Run the updated application after the update.	The application runs successfully after the update.

3.3 File system control

Table 3. Testing the file system control

Description	Expected result
Working with the file system without restrictions. Checking how the policies work.	
1) Create a file system rule for the *No_rules* directory and its top-level files and subfolders (*C:\No_rules\#*#*), with Read, Write and Delete operations allowed. Be sure to tick off **Active** for the rule.	If the directory opens successfully, then Read operation has succeeded. If the file is created successfully, Write op-

Description	Expected result
2) Open the directory with Windows Explorer. 3) Create a new file in the directory. 4) Remove the file you just created.	eration has succeeded. If the file is removed successfully, Delete operation has succeeded.
Working with the file system with the Read and Delete restrictions for all applications	
1) Create a new file (*newfile.txt*) in the *No_write_delete_all* directory. Create a rule for the *No_write_delete_all* directory and its top-level files and subfolders (*C:\No_write_delete_all\#*#*), with Read operation allowed and Write and Delete operations blocked. Apply the rule to all applications. Be sure to tick off **Active** for the rule. 2) Open the directory with Windows Explorer. 3) Try to create one more file in it. 4) Try to remove *newfile.txt*.	If the directory opens successfully, then Read operation has succeeded. If the file cannot be created, Write operation is blocked. If the file cannot be removed, Delete operation is blocked.
Working with the file system with the Read restriction for a certain application Local settings only	
1) Create a custom file system rule in the properties of *FAR.exe* (or of the application that you use instead of it), for the *No_write_program* directory and its top-level files and subfolders (*C:\No_write_program \#*#*), with Read and Delete operations allowed and Write operation blocked. Be sure to tick off **Active** for the rule. 2) Open the directory with Windows Explorer. Create a new file in the directory. 3) Open the directory with *FAR.exe*. Create a new file in the directory.	If the directory opens successfully with *Explorer.exe*, then Read operation has succeeded. If the file is created successfully with *Explorer.exe*, Write operation has succeeded. If the directory opens successfully with *FAR.exe*, then Read operation has succeeded. If the file cannot be created with *FAR.exe*, Write operation is blocked.
Working with the file system with restrictions for a certain user Local settings only	
1) Create a rule for the *Not_for_you* directory and its top-level files and subfolders (*C:\Not_for_you\#*#*), for the **Trusted applications** zone, with Read operation allowed and Write and Delete operations blocked. Be sure to tick off **Active** for the rule. In the Additional properties, apply the rule only to *USER2*. 2) Perform the following operations under the *USER1* account: • open the directory with Windows Explorer; • create a new file in the directory; • remove the file you just created. 3) Perform the following operations under the *USER2* account: • open the directory with Windows Explorer;	If *USER1* opens the directory successfully, then Read operation has succeeded. If *USER1* creates the file successfully, Write operation has succeeded. If *USER1* deletes the file successfully, Delete operation has succeeded. If *USER2* opens the directory successfully, Read operation has succeeded. If *USER2* cannot create the file, Write operation is blocked.

Description	Expected result
• create a new file in the directory with the help of *FAR.exe*.	
Working with the file system with time restrictions	
1) Create a new file (*newfile.txt*) in the *Not_now* directory. Create a rule for the *Not_now* directory and its top-level files and subfolders (*C:\Not_now\#*#*), for the **Trusted applications** zone, with Read, Write and Delete operations blocked. Be sure to tick off **Active** for the rule. In the Additional properties, specify time interval so that the rule is valid right now but becomes invalid in several minutes. 2) Open the directory with Windows Explorer. Create a new file in it with the help of *FAR.exe*. 3) Wait until the rule expires. 4) Open the directory with Windows Explorer. Create a new file in it. Remove the file you just created.	If the directory does not open while the rule is valid, then Read operation is blocked. If the file cannot be created while the rule is valid, Write operation is blocked. If the directory opens successfully after the rule expires, Read operation has succeeded.

3.4 Registry control

To perform the test, you need to have administrator privileges on the client computer, or simply rights to modify the registry. The registry root keys in the path specified in the rules should be assigned as follows.

Table 4. Specifying the paths in the registry

Registry key	Assigned in the SoftControl SysWatch rules as
HKEY_CLASSES_ROOT	*\REGISTRY\MACHINE\SOFTWARE\CLASSES*
HKEY_LOCAL_MACHINE	*\REGISTRY\MACHINE*
HKEY_CURRENT_USER	*\REGISTRY\USER\<SID>* for the user with the specified security identifier (*<SID>*)
HKEY_USERS	*\REGISTRY\USER*

Table 5. Testing the registry monitoring

Description	Expected result
Working with the registry with restrictions for all applications	
1) Create the *HKLM\SYSTEM\Key1* key. Create a rule for the key: the path is *\REGISTRY\MACHINE\SYSTEM\KEY1#**#*, Write operation is blocked and Delete operation is allowed. 2) Run *Regedit.exe* and try to create a subkey and to remove *Key1*.	Creating subkeys is blocked. Removing *Key1* is allowed.
Working with the registry with restrictions for a certain user Local settings only	
1) Create the *HKLM\SYSTEM\Key2* key. Create a rule for the key: the path is	All operations are completed suc-

Description	Expected result
*\REGISTRY\MACHINE\SYSTEM\KEY2#**#* and Write and Delete operations are blocked. In the Additional properties, apply the rule to *USER2* only. Note: it is essential that both *USER1* and *USER2* have the rights to run *Regedit.exe*; for example, they can be administrators. 2) Run *Regedit.exe* under the *USER1* account. Try to rename *Key2*, to create a subkey or a parameter, and to remove *Key2*. 3) Perform the same operations under the *USER2* account.	cessfully for *USER1*. All operations are blocked for *USER2*.
Working with the registry with time restrictions	
1) Create the *HKLM\SYSTEM\Key3* key. Create a rule for the key: the path is *\REGISTRY\MACHINE\SYSTEM\KEY3#**#* and Write and Delete operations are blocked. In the Additional properties, specify time interval so that the rule is valid right now but becomes invalid in several minutes. 2) Run *Regedit.exe*. Try to rename *Key3*, to create a subkey or a parameter and to remove *Key3*. 3) Wait until the rule expires. 4) Run *Regedit.exe* again and perform the same operations.	All operations are blocked while the rule is valid. After the rule expires, all operations are performed successfully.

3.5 USB drive control

Table 6. Testing USB drive control

Description	Expected result
Blocking access to all USB drives	
1) Deselect **Read**, **Write** and **Delete** for **USB devices**. 2) Connect *STOR1* to a USB port and perform Read, Write and Delete operations for the files on the drive. 3) Connect *STOR2* to a USB port and perform Read, Write and Delete operations.	All operations with the file system on *STOR1* and *STOR2* are blocked.
Allowing access to certain USB drives	
1) Connect *STOR1* to a USB port. In the Additional properties of the rule, add the drive to the exceptions and allow Read, Write and Delete operations. Perform Read, Write and Delete operations for the files on *STOR1*. 1) Disconnect *STOR1* and connect *STOR2* to a USB port. 2) Perform Read, Write and Delete operations for the files on *STOR2*. 3) Disconnect *STOR2* and connect *STOR1* to a USB port.	All operations with the file system on *STOR1* are allowed. All operations with the file system on *STOR2* are blocked.

Description	Expected result
4) Perform Read, Write and Delete operations for the files on *STOR1*.	

4. Customer support

If you have any questions concerning the installation, setting up and operation of SoftControl SysWatch, please contact our customer support by e-mail support@safensoft.com.