RE:Crypt

System Administrator Setup and Operations Guide for ReHIPS 1.2.0 Host-based Intrusion Prevention System

07.28.2014

RE:Crypt

Table of Contents

Technology

Basic Security Mechanisms of the Windows Operating System

All access to objects (processes, files, registry keys, etc.) in Windows is implemented via the security reference monitor (SRM). On each access the SRM checks whether the subject has sufficient rights to access the object. This involves two entities: an *access token* and a *security descriptor*.

Access Token

An access token contains information about the subject. In particular, it contains a list of *security identifiers* (SIDs), which include *mandatory label* and a list of privileges.

In the access token there is one primary SID and any number of additional identifiers. In other words, the access token corresponds to one main subject, who may be a member of different groups, e.g. a group of subjects of a single logon session (Logon SID), the «Everyone» group, a group of the specific integrity level, the «Users» group, the «Authenticated Users» group and others.

Security Descriptor

A security descriptor contains information about the object. It includes two access control lists (ACLs): *discretionary* (DACL) and *system* (SACL).

Security Identifier

A security identifier is a unique identifier within a single machine, which identifies the subject. Subjects in this case may be different: users, groups of users (e.g. Administrators), as well as some abstract entities like subjects of a single logon session (Logon SID).

Mandatory Label

The mandatory access control model is implemented using mandatory labels. Access is granted only when the integrity level of a subject is not lower than such of an object. In this case only access attempts subjected to the following policies are checked. Three policies are in place:

- restricts reading of more privileged objects (no read-up); by default applies to reading the process address space;
- restricts writing to more privileged objects (no write-up); by default applies to every object;
- restricts execution of more privileged objects (no execute-up); by default applies to the execution of more privileged COM objects.

Privileges

Privileges are the rights of a subject to perform various actions related to the operating system (such as loading drivers, shutting down the system and so on). Privileges can be present or absent in the access token. When they are absent there is no way to add them, but present privileges can be removed. If a privilege is present in the access token, it can be enabled and disabled. Some privileges are enabled by Windows if necessary; some must be enabled manually.

Discretionary Access Control List

The discretionary access control model is implemented using discretionary access control lists. For each object there is a list of entries. Each entry specifies access rights allowed or denied for a subject. If an object does not have a DACL, full access is granted to all subjects. If the list is present but empty, all access attempts to the object are denied.

Order of the entries does matter. When access is requested, the entries are examined until one of the following conditions is met: all the requested access rights can be granted (access will be granted), at least one of the requested rights denied (access will be denied) or the end of the list is reached (access will be denied).

System Access Control List

System access control list contains information about object auditing. Each entry specifies the types of access attempts by an object that cause a record in the security event log to be generated. It also contains the mandatory access control label.

Integrity Levels

In addition to the discretionary access control Windows has the mandatory access control. It is implemented using mandatory access control labels called integrity levels.

Access Rights Inheritance

If an object is a container that can contain other objects (e.g. files in a folder), access rights can be inherited by child objects if it was specified in access rights of the parent object.

Default Access Rights

A security descriptor can be specified at object creation. If the security descriptor is not specified, default access rights are assigned to the object. Hereby the integrity level for some objects is taken from the access token of a creating subject and for some objects it is not explicitly assigned (in that case an implicit medium integrity level is applied). Regarding the discretionary access, system, administrators and the creator of the object are granted full access to the object. The object can also inherit additional access rights from its parent.

User Interface Privilege Isolation

User interface privilege isolation is implemented on the basis of integrity levels. An application cannot interact with windows of an application with a higher integrity level. Sending of some window messages is meant by interaction (basically those leading to writing, for example WM_SETTEXT; reading messages, such as WM_GETTEXT, will be delivered successfully) as well as setting of window hooks, injection of a dynamic link library (DLL), etc.

How Windows Security Mechanisms Work

Consider a situation in which a running application intends to access an object such as file. At first the mandatory access control takes place. The application integrity level is checked against required integrity level to access the object requested. If the level is not high enough, the application will not be able to write to the file, but may try to access it for reading. If the application integrity level is higher or equal to the requested object integrity level, the application can write to it.

If the application has successfully passed the mandatory access control, it is the discretionary access control turn. The security reference monitor examines the object (the file in this case) discretionary access control list and grants or denies access.

By default a user has write access to almost all files and directories belonging to him. He also has read and execute access rights to all files in all folders except for the directories and files of other users. Moreover by default a user has write access right to all directories except system ones and directories of other users.

Desktop is a Security Boundary

If two applications with the same integrity level were started on the same desktop and DESKTOP_HOOKCONTROL access right was set for the first one, then the first application would be able to set window hooks on the second one's windows and possibly execute arbitrary code in the context of the second application.

If DESKTOP_HOOKCONTROL access right was not set, then the application may work incorrectly because runtime libraries use window hooks quite often. And even without this access right it's still possible to perform some unwanted actions like taking screenshot of the current desktop.

ReHIPS allows starting a restricted application on a separate desktop. Being run on a separate desktop restricted applications can set any window hooks they want and take screenshots while other applications are safe and secure.

ReHIPS as an Extension of the Windows Security Subsystem

The host-based intrusion prevention system ReHIPS allows running applications in one of two modes: unrestricted mode and ReHIPS mode.

Unrestricted mode allows applications to function without any restrictions imposed by ReHIPS.

Restrictive mode (ReHIPS mode) is a mode, in which the application runs on behalf of the user, which was created by ReHIPS specifically for this application (the so-called ReHIPS-user). Access to other objects (which do not belong to a ReHIPS-user) will be allowed only if the security descriptors of these objects have allowance entries for the ReHIPS-user. A number of ReHIPS-users and applications which run on their behalf are configured by system administrator.

This means that every potentially dangerous application will run on behalf of its own ReHIPS-user, which allows the administrator to restrict access to the sensitive operating system objects. Any attempts to violate policies will be stopped by the Windows security subsystem. Even if any malicious code runs, its ability to access operating system objects will be bound by the system administrator settings.

All information about applications is stored in a database, which is represented by a file in XML format. Access to the file is allowed only to the operating system and the administrators group, which blocks any unauthorized access. ReHIPS works directly with the database file, which allows maintaining up-to-date information and takes into account any user changes to it. All write operations use transactions with backup, which provide integrity of the database even if an unexpected application shutdown occurs while writing to the file.

Running an Application

When an application is started the following occurs.

If the user chooses the «Allow restricted» option on the first start of the application, the execution is blocked and the application is terminated. Then ReHIPS-user with the specified security settings (access rights, privileges, etc.) is used for this application. After that the application is restarted on behalf of this user.

If the user chooses the «Allow» option, the application will be started without any restrictions.

Applications Installation Using DeployHelper

DeployHelper is a technology designed to simplify installation of new software into the system protected by ReHIPS. It is based on the fact that applications installed should be run on behalf of the user running the installer. Otherwise the applications may have no access to the user settings, created during the installation process, which may lead to incorrect work of the installed software. Thus it is not recommended to run the installer in ReHIPS mode.

If the installer runs in unrestricted mode, the installation will be on behalf of the current user. According to the ReHIPS principles, application execution will be on behalf of another user (ReHIPS-user), which may also lead to incorrect work of the installed software. In this case it may be necessary to configure ReHIPS and application environment manually.

To install an application using DeployHelper one needs to click the right mouse button on the application installer in the Explorer and select the «Run in ReHIPS DeployHelper» menu item. Either choose «Run as administrator in ReHIPS DeployHelper» if the installer needs administrator rights. During the installation

process the executable file browse dialogue will be shown. Every selected executable file will be added to the ReHIPS database with correct settings. Being installed that way the software can be executed in ReHIPS mode.

ReHIPS Setup

During initial ReHIPS setup RulesPack32.exe (or RulesPack64.exe on 64-bit Windows) is executed, which adds applications with predefined rules to the database. An application is added to the database only if:

- it was installed before ReHIPS;
- a corresponding predefined rule is found in RulesPack;
- it is not found in the ReHIPS applications database.

RulesPack

RulesPack application contains predefined rules for some widespread applications. If a separate folder is necessary for an application (for instance, WinWord works with files), this folder is created at <system disk>:\ReHIPS\<application-specific folder>. All the necessary access rights to this folder are granted. Some applications may keep their settings in user's home directory or in user's registry keys. These applications being run in ReHIPS mode will not be able to access them. So all the necessary registry keys are copied to the registry of respective ReHIPS-user and all the necessary directories are symlinked at. In both cases all the necessary access rights to these objects are granted.

Predefined rules are constantly updated. Yet some applications may still be missing. In that case ReHIPS manual setup is required.

RulesPack can be manually executed. To do it one should find the folder ReHIPS was installed in (<system disk>:\Program Files\ReCrypt\ReHIPS by default) and run RulesPack32.exe (or RulesPack64.exe on 64-bit Windows) application.

ReHIPS Manual Setup

ReHIPS manual setup involves applications editing to set applications access rights to different operating system objects.

To edit an application one should find and double-click it in the applications list in the ReHIPS main window (fig. 1) or click the Edit button (marked with the red frame).

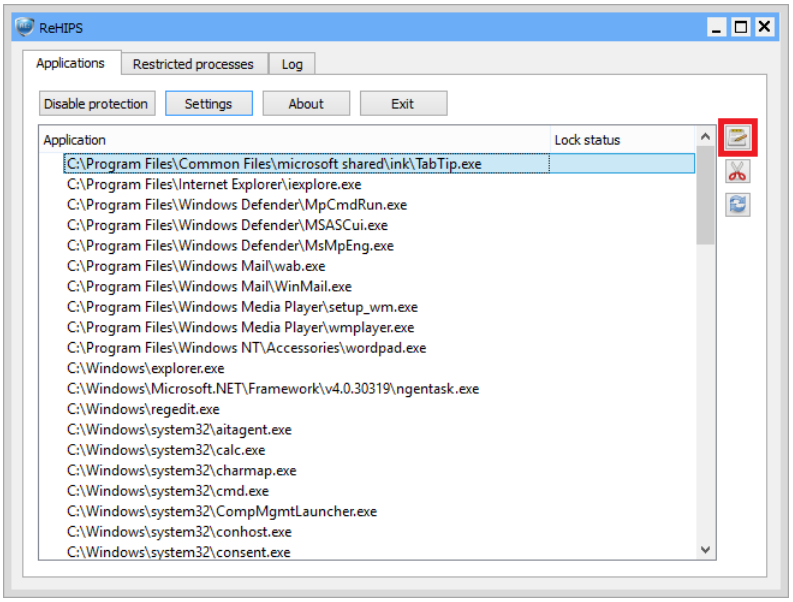

Fig. 1 — ReHIPS main window

The following edit application window will appear (fig. 2).

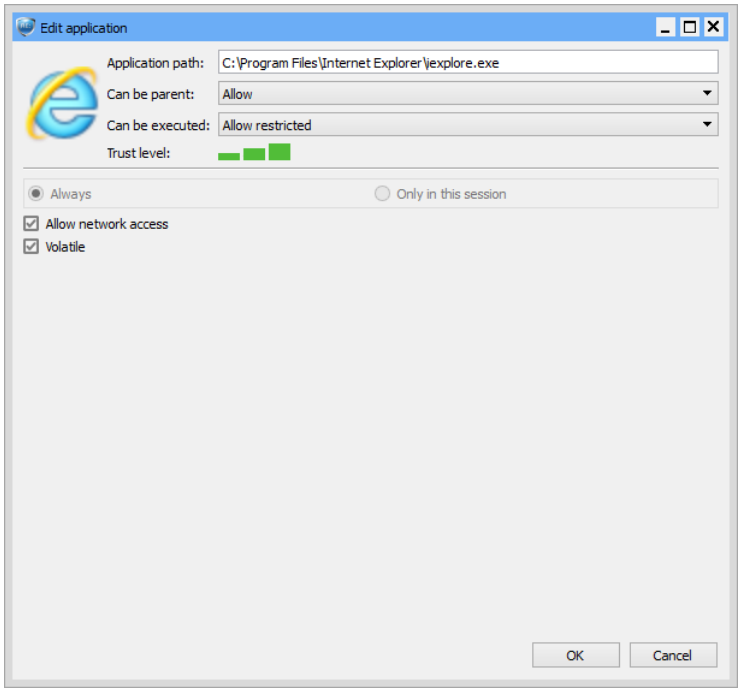

Fig. 2 — Edit application window

A window similar to fig. 2 is also shown for any application being run that is not found in the ReHIPS database. A similar window appears for parent processes.

The color of the Trust level indicator is green. It means the application file is Microsoft Authenticode signed and the signature is valid. Yellow color denotes unsigned application file. Red indicator stands for signed

application file that failed signature verification. Grey color denotes the inability to determine trust level, e.g. due to some file system internals.

Modes of Execution

There are four possible options in «Can be parent» drop-down list (fig. 3).

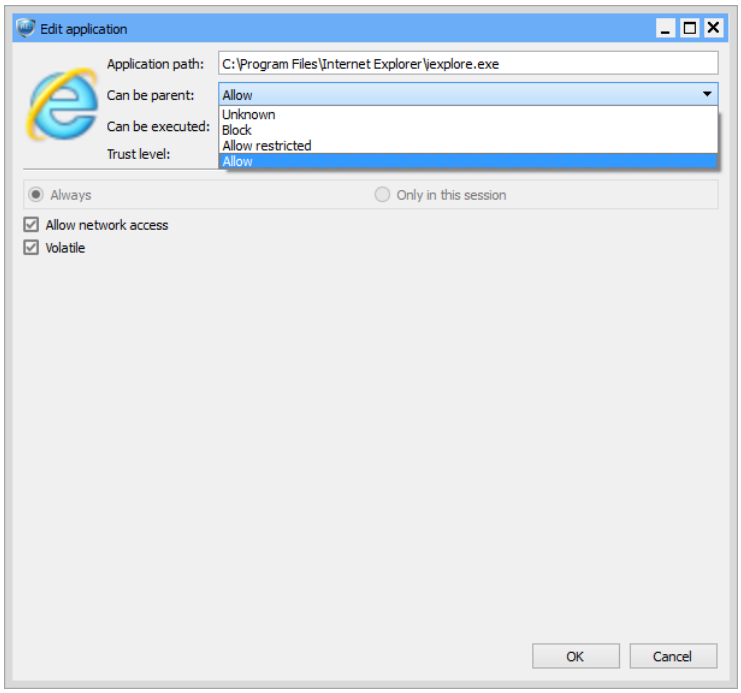

Fig. 3 — «Can be parent» drop-down list in Edit application window

This setting defines whether the application can create child processes. Let's take a closer look at possible options.

- **Unknown**. With this option in effect an attempt to create a child process will result in showing the ReHIPS window asking whether this action should be allowed.
- **Block**. With this option chosen an attempt to create any child process will fail.
- **Allow restricted**. With this option selected an attempt to create a child process will cause inspection of the child application whether it's allowed to execute. May be used for instance with Explorer, that can run either untrusted applications in ReHIPS mode or trusted ones in unrestricted mode.
- **Allow**. With this option set an attempt to create a child process will succeed, the child process will inherit all the rights and privileges of its parent. May be used when the child and parent processes are somehow connected. For instance Internet Explorer spawns several processes to implement separate processing of different tabs.

There are four possible options in «Can be executed» drop-down list (fig. 4).

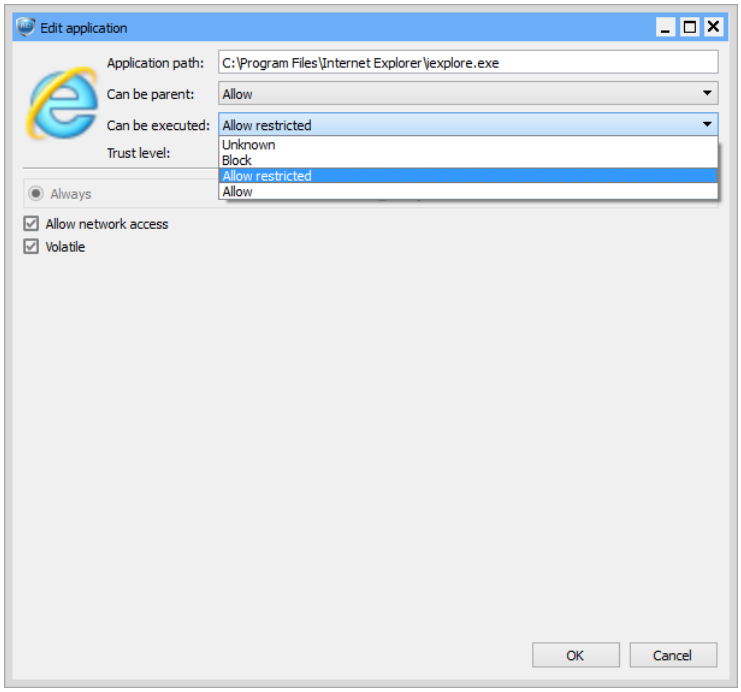

Fig. 4 — «Can be executed» drop-down list in Edit application window

Let's discuss them in detail.

- **Unknown**. With this option in effect an attempt to run the application will result in showing the ReHIPS window asking whether this application should be allowed to execute.
- **Block**. With this option chosen an attempt to run the application will fail.
- **Allow restricted**. With this option selected an attempt to run the application will end up in application execution in ReHIPS mode.
- **Allow**. With this option set an attempt to run the application will succeed without any restrictions.

The application network access can be allowed by checking «Allow network access» checkbox.

«Volatile» checkbox can be checked to suppress warning window shown (fig 5) when the application executable file changes.

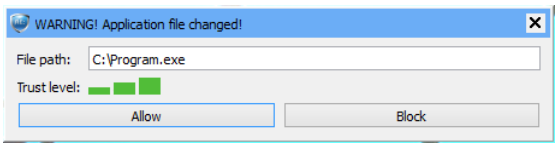

Fig.5 — Changed application file warning window

«Only in this session» option denotes whether all these settings will be kept during current session only (application settings will be kept in the database during ReHIPS current session and will be discarded on reboot or ReHIPS service stop).

If some application being run is not found in the ReHIPS database the following window will be shown asking whether this application should be allowed to execute (fig. 6):

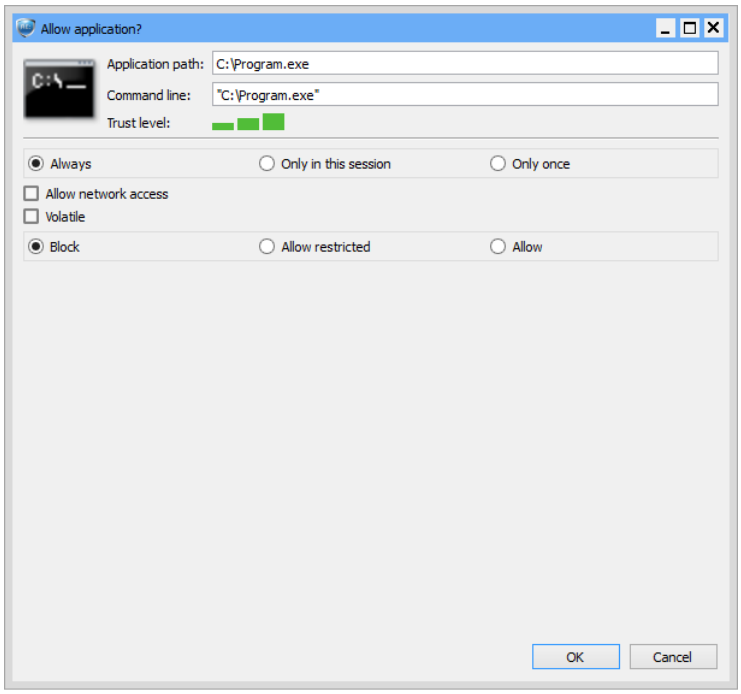

Fig. 6 — Allow application window for a new application

Three modes of execution are available:

- **Allow** — application execution will be allowed without any restrictions;
- **Allow restricted** — application will be run in ReHIPS mode;
- **Block** — application execution will be blocked.

By checking respective checkboxes it's also possible to allow network access, mark file as volatile or to keep all these settings only during current session or not to keep them at all. Keep in mind that with «Only in this session» or «Only once» chosen application execution can be allowed or blocked only.

After the first execution and setup application settings are saved in the ReHIPS database (with «Only in this session» and «Only once» unchecked) and its further customization can be done by editing from the main ReHIPS window.

A similar window appears for parent processes.

Advanced Mode

To enable advanced mode press «Settings» button in the main ReHIPS window (fig. 7) and check «Advanced mode» in the settings window (fig. 8).

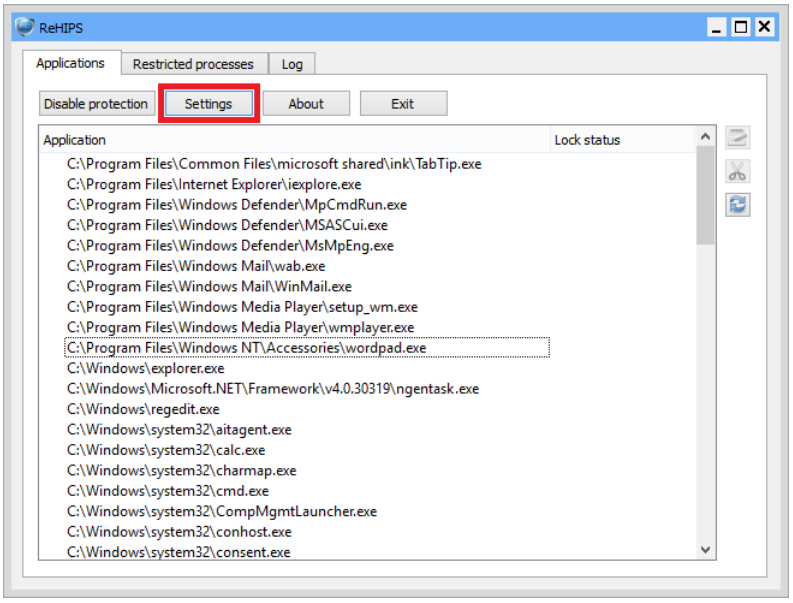

Fig. 7 — «Settings» button in the main window

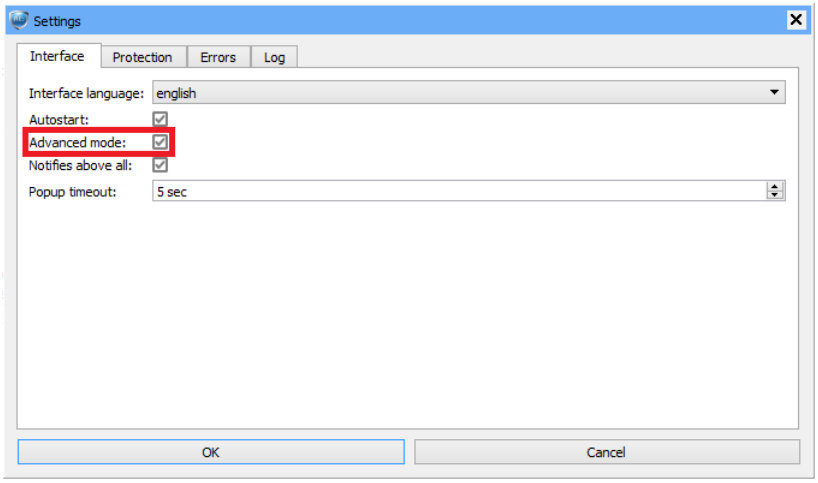

Fig. 8 — Advanced mode in settings window

Allow application window for a new application in advanced mode looks as follows (fig. 9):

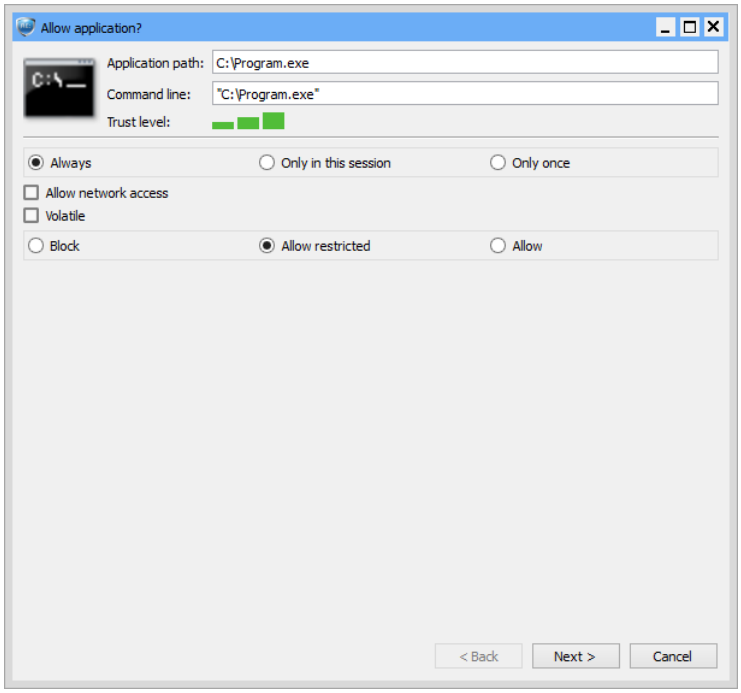

Fig. 9 — Allow application window for a new application in advanced mode

In advanced mode it becomes possible to configure a user when "Allow restricted" is checked. When "Next" button is pressed the wizard proceeds to the Edit new user window (fig. 10). It is also possible to run an application on behalf of already existing user (fig. 11).

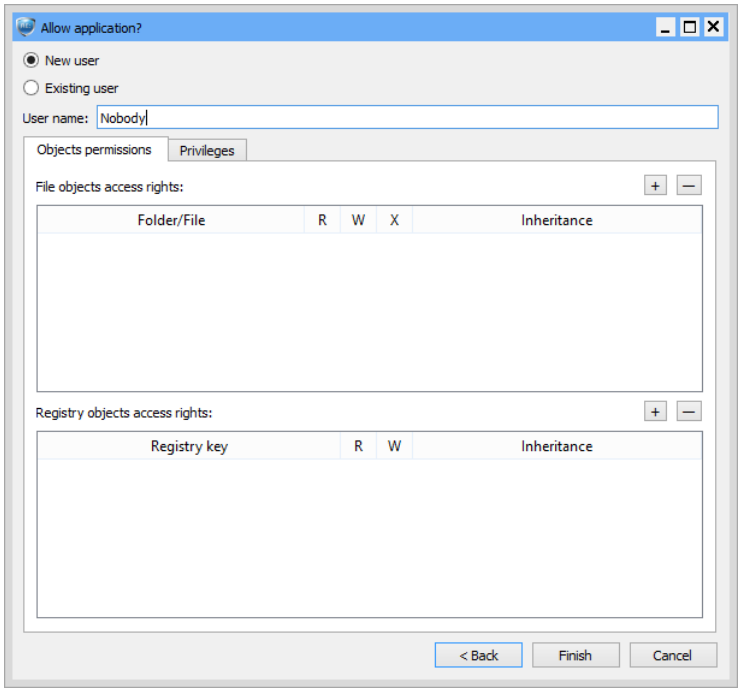

Fig. 10 — Edit new user window

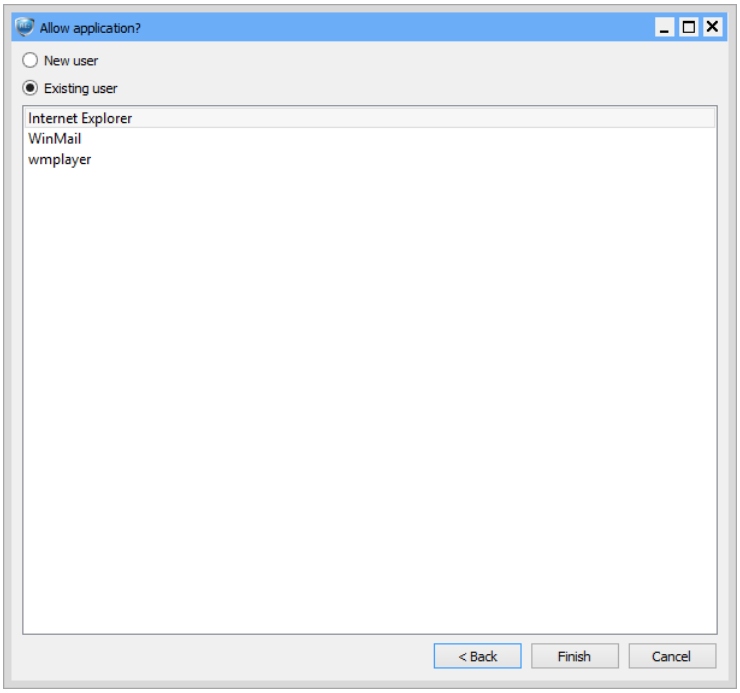

Fig. 11 — Existing user selection window

User editing becomes available from the main window in advanced mode (fig. 12). To edit a user one should find and double-click it in the applications list in the ReHIPS main window or click the Edit button (marked with the red frame).

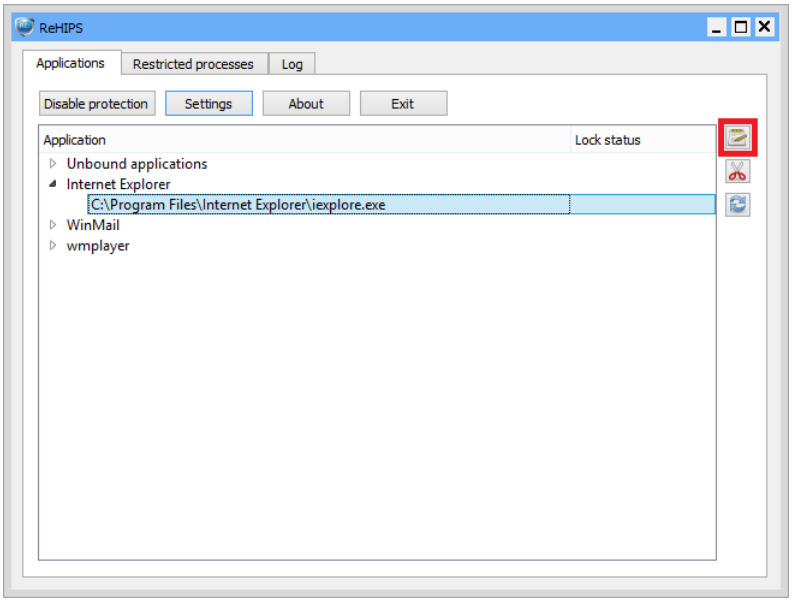

Fig. 12 — Main window in advanced mode

The “Unbound applications” group here denotes applications, which have never been run in restricted mode and no user was created for them.

Edit application window in advanced mode looks as follows (fig. 13):

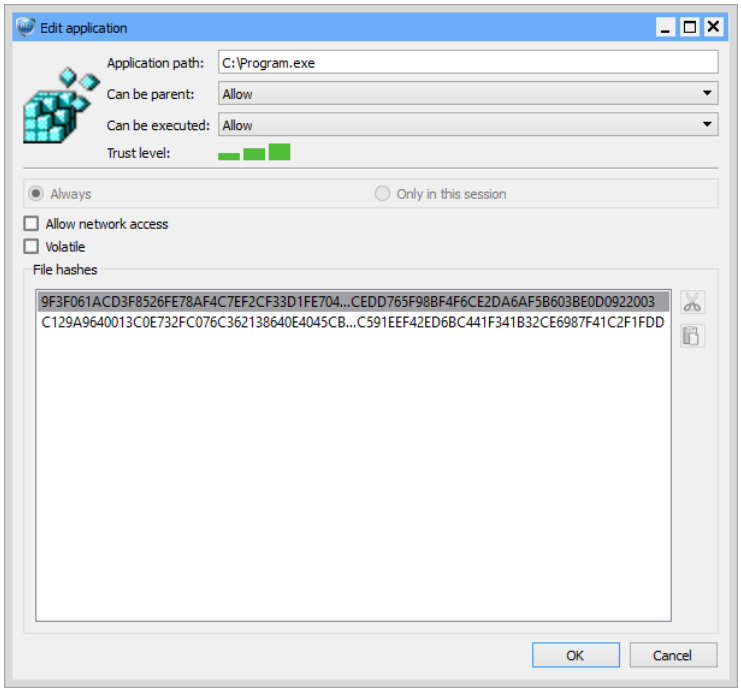

Fig. 13 — Edit application window in advanced mode

Advanced mode enables to manage file hashes. You can remove hashes or copy them to clipboard. Actual hash of the application file is marked with grey. When file of an application being run changes, the following window appears in advanced mode (fig. 14):

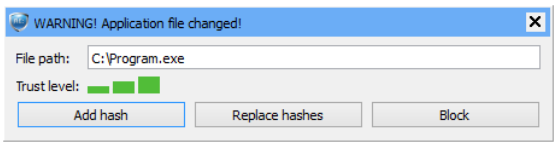

Fig. 14 — Changed application file warning window in advanced mode

When “Add hash” button is pressed a hash list will be appended with the actual hash. When “Replace hashes” button is pressed the hash list will be replaced with the actual hash.

Objects Permissions

Objects in ReHIPS are comprised of file objects (files and folders) and registry objects (keys). Access rights to the file objects contain read, write and execute rights (R, W and X respectively). Registry objects access rights include read and write rights (R, W).

Green and red checkboxes can be seen at fig. 15. Green checkboxes allow respective access to the object. Red ones deny.

You can add or delete objects using «+» and «-» buttons.

«Inheritance» column defines inheritance of the chosen access rights. Drop-down list (fig. 15) consists of 4 options.

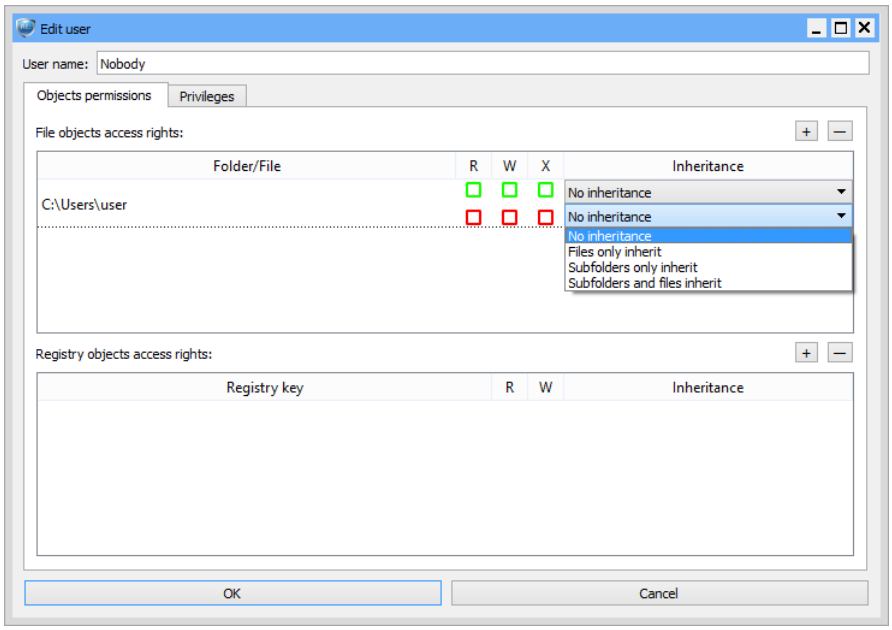

Fig. 15 — «Inheritance» drop-down list in Edit user window

Let's take a closer look at these options.

- **«No inheritance»** denotes no access rights inheritance, all access rights are applied to the file object only.
- **«Files only inherit»** stands for access rights inheritance for files in the folder if the chosen file object is a folder (but subfolders do not inherit any access rights).
- **«Subfolders only inherit»** leads to access rights inheritance for subfolders in the folder if the chosen file object is a folder (but files do not inherit any access rights).
- **«Subfolders and files inherit»** means access rights inheritance for subfolders and files in the folder if the chosen file object is a folder. In other words, all child objects inherit access rights.

The same applies to the registry objects.

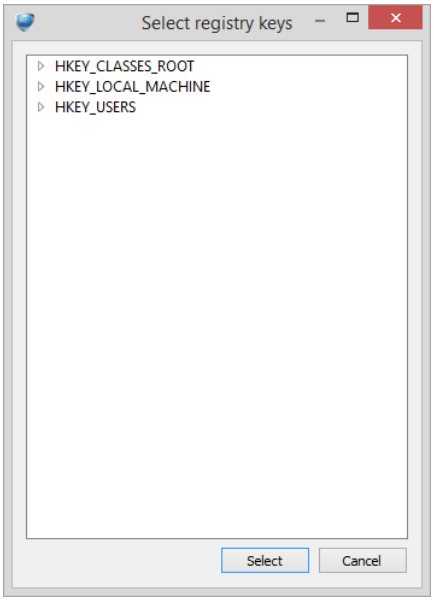

Fig. 16 — Registry keys selection window

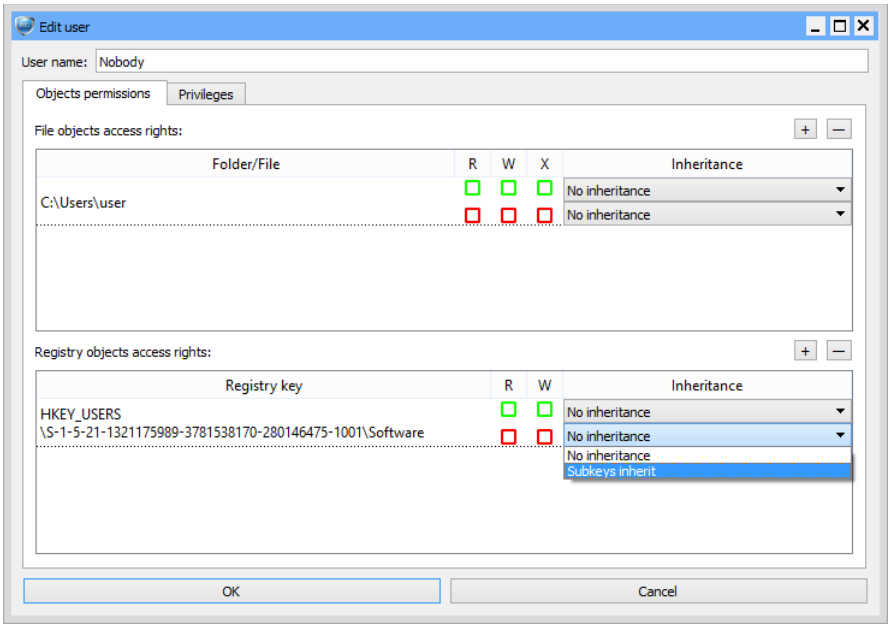

Fig. 17 — Registry objects access rights in Edit user window

When «+» button for the registry objects is pressed, registry keys selection window will be shown (fig. 16). Edit user window with added registry object is shown at fig. 17.

Privileges

«Privileges» tab in Edit user window contains user privileges and access rights.

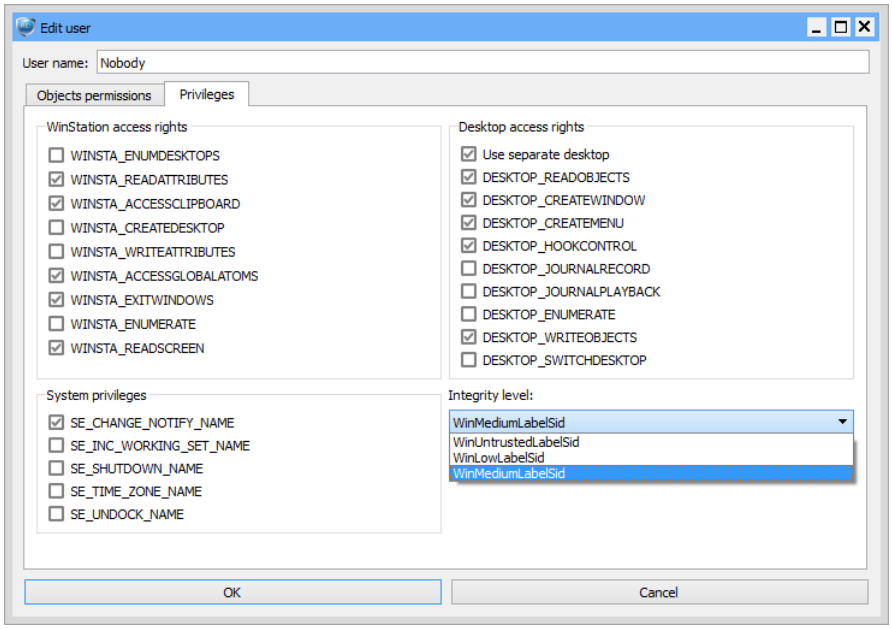

Fig. 18 — «Privileges» tab in Edit user window

Fig. 18 shows possible user privileges and access rights. Let's take a closer look at them.

WinStation Access Rights

Access right	**Description**
WINSTA_ENUMDESKTOPS	allows to enumerate existing desktop objects
WINSTA_READATTRIBUTES	allows to read the attributes of a window station object including color settings and other global window station properties
WINSTA_ACCESSCLIPBOARD	allows to use the clipboard
WINSTA_CREATEDESKTOP	allows to create new desktop objects on the window station
WINSTA_WRITEATTRIBUTES	allows to modify the attributes of a window station object including color settings and other global window station properties
WINSTA_ACCESSGLOBALATOMS	allows to manipulate global atoms
WINSTA_EXITWINDOWS	allows to successfully call the ExitWindows or ExitWindowsEx function
WINSTA_ENUMERATE	allows for the window station to be enumerated
WINSTA_READSCREEN	allows to access screen contents

WINSTA_ENUMDESKTOPS allows to enumerate existing desktop objects that allow **DESKTOP_ENUMERATE** access right, security impact is minimal, unchecking is possible.

WINSTA_READATTRIBUTES allows to read the attributes of a window station object including color settings and other global window station properties, security impact is minimal, this access right is required to get cursor position, recommended to check, unchecking results in incorrect work of almost all applications.

WINSTA_ACCESSCLIPBOARD allows to use the clipboard, security impact is medium, there is some risk, but recommended to check, unchecking results in failure of clipboard operations.

WINSTA_CREATEDESKTOP allows to create new desktop objects on the window station, security impact is medium as an application may display fraudulent data on a new desktop, unchecking is possible.

WINSTA_WRITEATTRIBUTES allows to modify the attributes of a window station object including color settings and other global window station properties, security impact is medium as it's undesired for untrusted applications to modify settings, unchecking is possible.

WINSTA_ACCESSGLOBALATOMS allows to manipulate global atoms, security impact is medium and similar to clipboard access right, recommended to check, unchecking results in failure to run almost all applications.

WINSTA_EXITWINDOWS allows to successfully call the ExitWindows or ExitWindowsEx function, security impact is minimal to medium as system shutdown may lead to denial of service, but **SE_SHUTDOWN_NAME** system privilege is also required for this to work, recommended to check, unchecking results in failure to run almost all applications.

WINSTA_ENUMERATE allows for the window station to be enumerated, security impact is minimal, unchecking is possible.

WINSTA_READSCREEN allows to access screen contents, security impact is medium, recommended to check, unchecking results in possible incorrect work of almost all applications.

Desktop Access Rights

Access right	Description
Use separate desktop	allows to start an application on a separate desktop
DESKTOP_READOBJECTS	allows to read objects on the desktop
DESKTOP_CREATEWINDOW	allows to create a window on the desktop
DESKTOP_CREATEMENU	allows to create a menu on the desktop
DESKTOP_HOOKCONTROL	allows to establish any of the window hooks
DESKTOP_JOURNALRECORD	allows to perform journal recording on a desktop
DESKTOP_JOURNALPLAYBACK	allows to perform journal playback on a desktop
DESKTOP_ENUMERATE	allows for the desktop to be enumerated
DESKTOP_WRITEOBJECTS	allows to write objects on the desktop
DESKTOP_SWITCHDESKTOP	allows to activate the desktop using the SwitchDesktop function

Use separate desktop option allows starting an application on a separate desktop. Click on a Desktop switching icon (fig 19) and then select desktop from the list of desktops screenshots (fig 20).

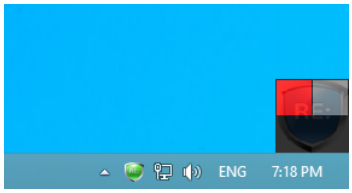

Fig. 19 — Desktop switching icon

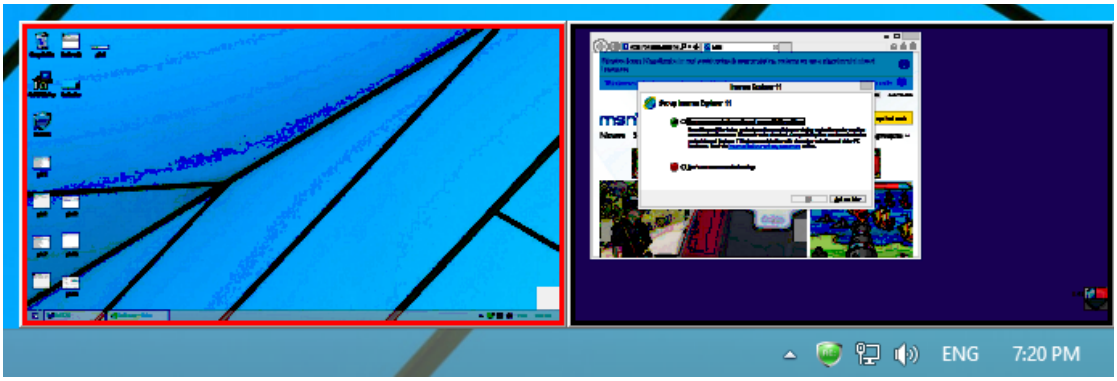

Fig. 20 — Switching between desktops

DESKTOP_READOBJECTS allows to read objects on the desktop, security impact is medium, recommended to check, unchecking results in failure to run almost all applications.

DESKTOP_CREATEWINDOW allows to create a window on the desktop, security impact is minimal, recommended to check, unchecking results in possible incorrect work of many applications.

DESKTOP_CREATEMENU allows to create a menu on the desktop, security impact is minimal, recommended to check, unchecking results in incorrect work of many applications.

DESKTOP_HOOKCONTROL allows to establish any of the window hooks, but recommended to check, unchecking results in possible incorrect work of some applications. Use **DESKTOP_HOOKCONTROL** only with **Use separate desktop** checked to decrease security impact from critical to minimal.

DESKTOP_JOURNALRECORD allows to perform journal recording on a desktop, security impact is medium, unchecking is possible.

DESKTOP_JOURNALPLAYBACK allows to perform journal playback on a desktop, security impact is medium, unchecking is possible.

DESKTOP_ENUMERATE allows for the desktop to be enumerated, security impact is minimal, unchecking is possible.

DESKTOP_WRITEOBJECTS allows to write objects on the desktop, security impact is medium, recommended to check, unchecking results in failure to run almost all applications.

DESKTOP_SWITCHDESKTOP allows to activate the desktop using the SwitchDesktop function, security impact is medium as an application may display fraudulent data on a new desktop, unchecking is possible.

System Privileges

Privilege	Enabled/disabled	Description
SE_CHANGE_NOTIFY_NAME	Enabled	allows to receive notifications of changes to files or directories, also causes the system to skip all traversal access checks
SE_INC_WORKING_SET_NAME	Disabled	allows to allocate more memory for applications that run in the context of users
SE_SHUTDOWN_NAME	Disabled	allows to shut down a local system
SE_TIME_ZONE_NAME	Disabled	allows to adjust the time zone associated with the computer's internal clock
SE_UNDOCK_NAME	Disabled	allows to undock a laptop

SE_CHANGE_NOTIFY_NAME allows to receive notifications of changes to files or directories, also causes the system to skip all traversal access checks, recommended to check, unchecking results in failure to run some applications. Besides security impact is minimal.

SE_INC_WORKING_SET_NAME allows to allocate more memory for applications that run in the context of users, security impact is minimal, high memory consumption may lead to denial of service, unchecking is possible.

SE_SHUTDOWN_NAME allows to shut down a local system, security impact is medium, system shutdown may lead to denial of service, unchecking is possible.

SE_TIME_ZONE_NAME allows to adjust the time zone associated with the computer's internal clock, security impact is minimal, unchecking is possible.

SE_UNDOCK_NAME allows to undock a laptop, security impact is minimal, unchecking is possible.

Integrity Levels

Integrity level	Description
SECURITY_MANDATORY_UNTRUSTED_RID	untrusted level, minimal possible level
SECURITY_MANDATORY_LOW_RID	low integrity level
SECURITY_MANDATORY_MEDIUM_RID	medium integrity level, used by default
SECURITY_MANDATORY_MEDIUM_PLUS_RID	medium high integrity level
SECURITY_MANDATORY_HIGH_RID	high integrity level
SECURITY_MANDATORY_SYSTEM_RID	system integrity level
SECURITY_MANDATORY_PROTECTED_PROCESS_RID	protected process integrity level

Integrity levels higher than **SECURITY_MANDATORY_MEDIUM_RID** are assigned to local services, administrators and privileged users (for instance, backup operators). There is no point in setting them

for restricted applications. If an application really needs it, it should be a trusted application and it should be allowed to run in unrestricted mode.

SECURITY_MANDATORY_MEDIUM_RID – standard integrity level for majority of applications, recommended to use by default.

SECURITY_MANDATORY_LOW_RID – low integrity level, some applications may operate incorrectly as they will not be able to access files and folders in user's settings. Some registry keys will also be inaccessible. But if an application does not need it, it'll work fine and it'll have higher security level.

SECURITY_MANDATORY_UNTRUSTED_RID – untrusted integrity level. Assigned by default on anonymous access, access to most objects is denied. Most likely applications will fail to work correctly, not recommended to use.

ReHIPS Settings

To customize settings press «Settings» button in the main ReHIPS window (fig. 7). The following settings window will appear (fig. 21).

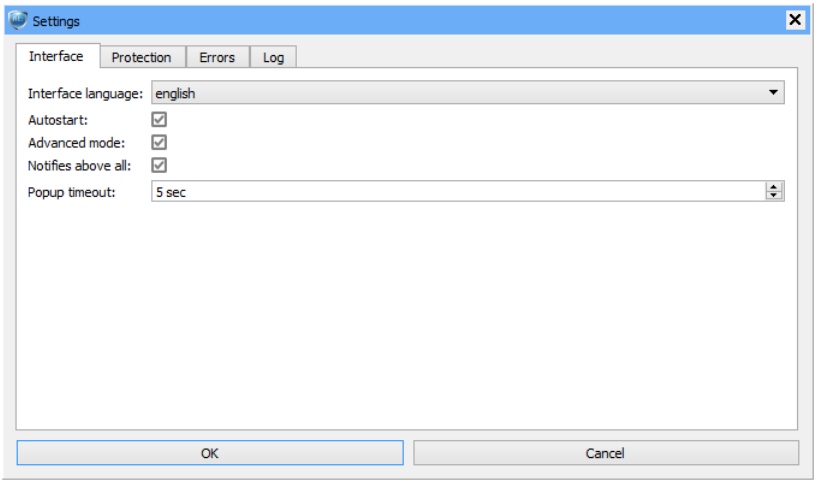

Fig. 21 — Interface settings window in advanced mode

Supported interface languages: English and Russian. Enabling advanced mode provides access to additional settings. If advanced mode is disabled in interface settings window, the last has the following look (fig. 22):

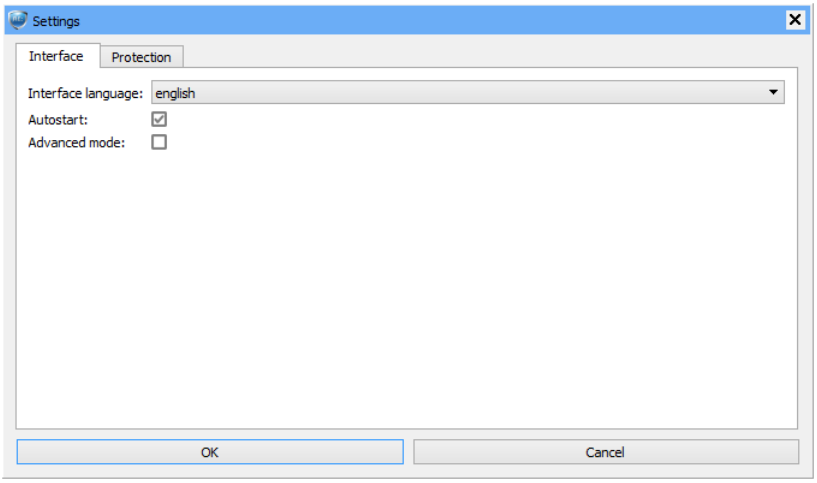

Fig. 22 — Interface settings window

If «Autostart» is checked ReHIPS will start automatically with the system. «Protection» tab of settings window has the following look (fig. 23):

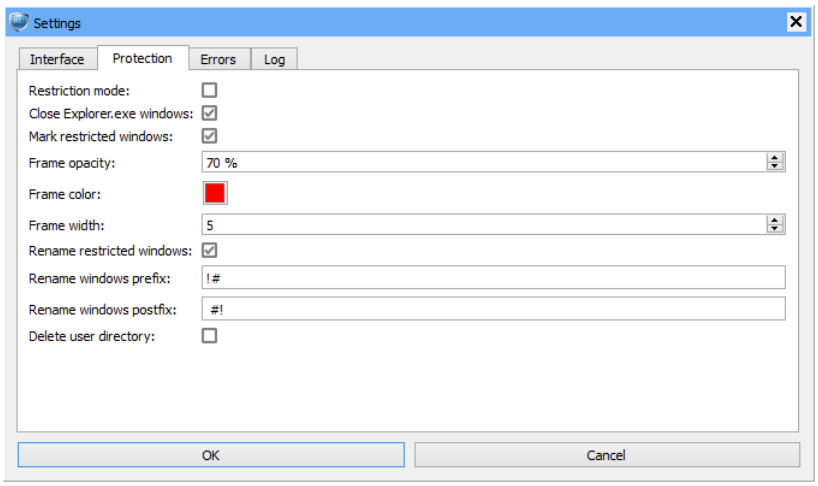

Fig. 23 — Protection settings window in advanced mode

If restriction mode is enabled ReHIPS works in a silent restriction mode blocking without asking everything changed and not explicitly allowed.

When an application is run in ReHIPS mode or when an application is blocked, Explorer.exe may show error window about failure to run the process. ReHIPS can automatically close these windows if «Close Explorer.exe windows» is checked.

If «Mark restricted windows» is checked, then windows of applications run in ReHIPS mode will be marked with a frame. Frame color will be taken from «Frame color» setting. If «Rename restricted windows» is checked, prefix and postfix from respective settings will be appended to the window caption.

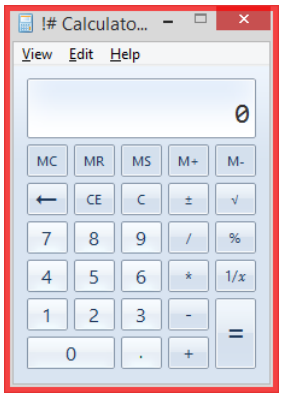

Fig. 24 — Window of the Calculator run in ReHIPS mode

«Delete user directory» setting controls deletion of user's home directory. When all the user's applications are deleted from the main ReHIPS window (fig. 12) respective ReHIPS-user is also removed. With this setting checked ReHIPS-user's home directory which may contain some useful files will be also deleted.

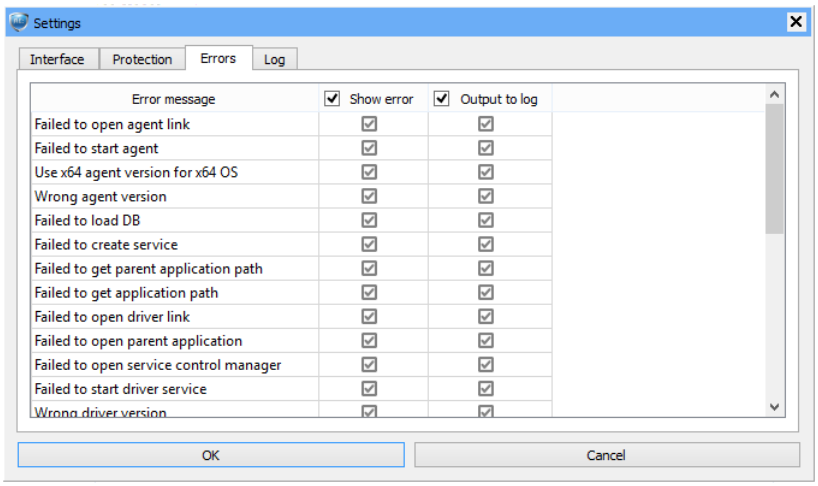

Fig. 25 — Errors settings window in advanced mode

«Errors» tab of settings in advanced mode (fig. 25) contains a list of errors ReHIPS can report. If «Show error» column is checked, the respective error will be reported of with a window (fig 26).

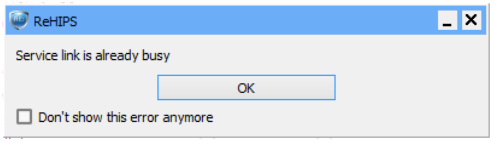

Fig. 26 — Error window

If «Output to log» is checked, the respective error will be logged.

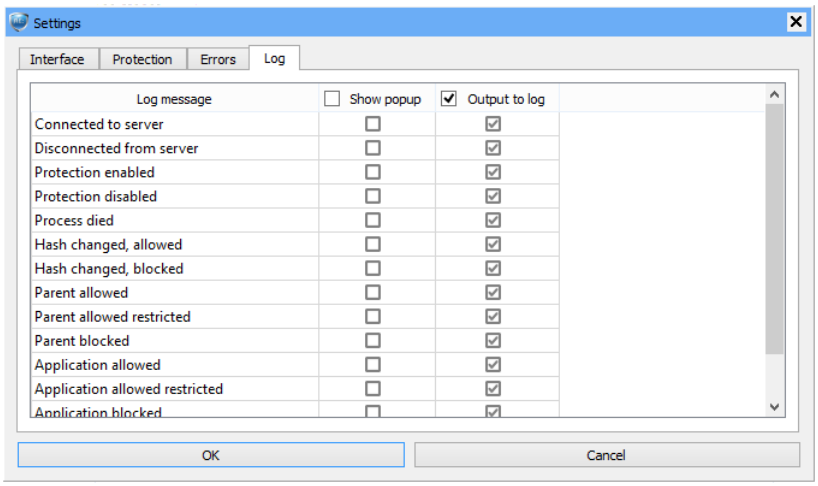

Fig. 27 — Log settings window in advanced mode

«Log» tab of settings in advanced mode (fig. 27) contains a list of events ReHIPS can log. If «Show popup» column is checked, the respective event will be reported of with a popup window (fig 28).

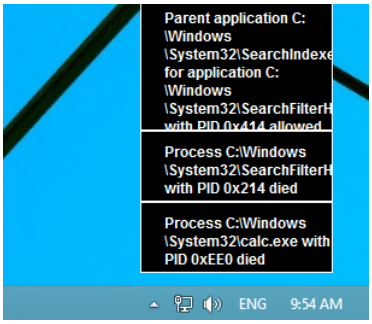

Fig. 28 — Popup windows

If «Output to log» is checked, the respective event will be logged.

ReHIPS log can be viewed at «Log» tab in the main ReHIPS window.

Operation features and Recommendations

Operating system already implements all the necessary operations to control and restrict access, so ReHIPS is not resource-demanding in operations.

Initial ReHIPS setup may take some time, if you grant minimal access rights to every application. But despite the restrictive ReHIPS policy (if it is not explicitly allowed, block it) thorough initial setup should be taken seriously as it provides higher security level.

Operation system vulnerabilities are one of the weak points. ReHIPS is based on Windows built-in security mechanisms, thus leading to vulnerabilities inheritance. But this should not be a big problem as operating system patches are released on a regular basis.

Another possible weak point is application vulnerabilities that lead to arbitrary code execution. In theory the executed code may perform some unwanted actions using application rights it was injected into (in Microsoft Word, for instance, it may delete some documents Word has access to).

Be warned that by default all users have read and execute access rights to all disks and authenticated users have write access right to all disks. These access rights are inherited by almost all the directory tree. In other words applications run in ReHIPS mode can read, write and execute almost any file from most of the directories excluding some special ones (like users' home directories). This leads to important recommendation: **all data should be stored only in user's home directory (on the desktop, in «My documents» directory, etc.)**. The same applies to applications executable files: **they should be stored in write-protected directories (in particular, Program Files)**.

ReHIPS including graphical user interface operates with Unicode thus being subjected to Unicode display features. In particular, the control symbol 0x202E (RLO – right-to-left override) will display the string from left to right. It's recommended to look through the following post http://blogs.technet.com/b/mmpc/archive/2011/08/10/can-we-believe-our-eyes.aspx about some Unicode display features. And it's also recommended to pay attention granting access rights to applications.