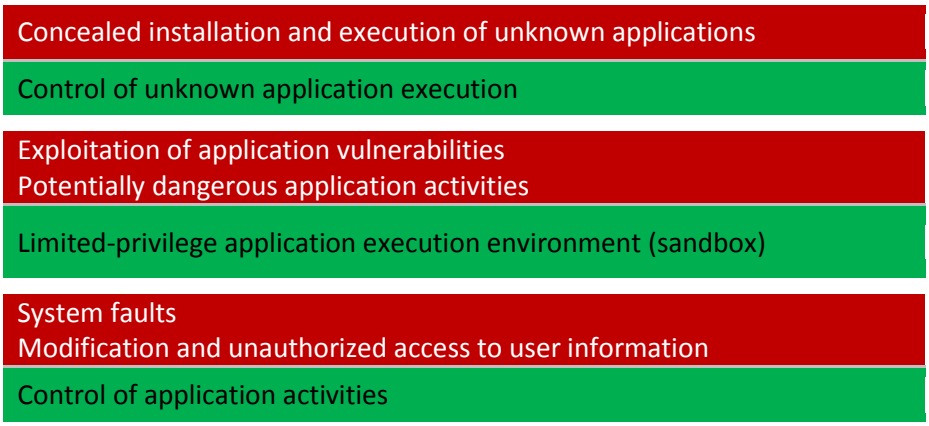
SafenSoft VIPO®

Valid Inside Permitted Operations Technology: a look under the hood



Components

- OS kernel mode drivers (Ring0)
- Operating system service
- User interface modules

Work cycle

System profile creation

The system profile is used to identify unknown applications and contains a list of all executable files (PE - Portable Executable) on the computer at the time of software installation. The PE file entry is the file's hash-sum calculated using the sha256 algorithm. To prevent malware from penetrating the system profile, each PE file is subjected to additional tests during the course of the system profile creation:

- Verification of authenticity
  If the file has a valid digital signature from a known developer or a root certificate from an authorized security center, it is added to the system profile. If not, the file undergoes further verification.
- System file verification
  If the file is registered in the Windows catalog (CATalog file), it is deemed to be an authentic Windows system file or installed program module and added to the System profile. If not, the file undergoes further verification.
- Virus scan
  If a file is a known virus, it is automatically excluded from the system profile. If the file is not a virus, it is added to the system profile. As it is possible that an unknown virus may be added to the system profile because it is too new to be detected by signature-based anti-virus, other means must be deployed to prevent this situation from causing future problems.

Control of unknown application execution

Whenever an executable module is loaded into system memory, its driver hash-sum is calculated and checked against the system profile. If the hash-sum exists in the system profile, the process data sent by the driver is accompanied by the *known* application parameter (Parameter.CreateProcess.Checked ==

TRUE), which confirms that the process hash-sum was checked for compliance with the system profile. Otherwise the process returns an “unknown application” result (Parameter.CreateProcess.Checked == FALSE). If an executable module such as a DLL, driver, etc., is loaded, its hash-sum is also checked in the system profile. If the hash-sum is not present, a request is sent to the TPSecure service to block the application.

Control of setup or update programs
TPSecure detects the execution of setup or update programs using the following indicators:

- File type information
  The file is a Microsoft Windows installer, Install Shield setup program.
- File version information
  The file version information contains combinations of the keywords setup, update, install.
- File digital signature
  The file has a valid digital signature from a known developer or a root certificate of an authorized security center

If an unknown application has a valid digital signature and version information, the application is designated an *authorized* setup or update program. On execution of an authorized setup program and application, the application itself and all files created by it during setup or update process are automatically registered in the system profile.

If an unknown application has no digital signature but does have valid file version and type information, the application is designated an *unauthorized* setup or update program. On execution of an unauthorized setup program and application, the application itself and all files created by it during setup or update process may be manually added to the system profile by an authorized user.

Limited-privilege application execution environment
The limited-privilege application execution environment is determined by the privileges of the user executing the application:

- User creation
  TPSecure creates a new VIPO user and grants/removes the relevant privileges for that user.
- Execution of applications on behalf of other users
  When an application is launched, TPSecure interrupts the execution of the application and launches it again on behalf of VIPO user.

The limitation of application privileges serves to block potentially dangerous application activities such as unauthorized driver setup, system shutdown, and integration with other applications.

Control of application activities
VIPO controls the following application activities:

- Execution and closing of application
- Use of system resources by the application
- Use of system registry resources by the application
- Establishment of network connection by the application