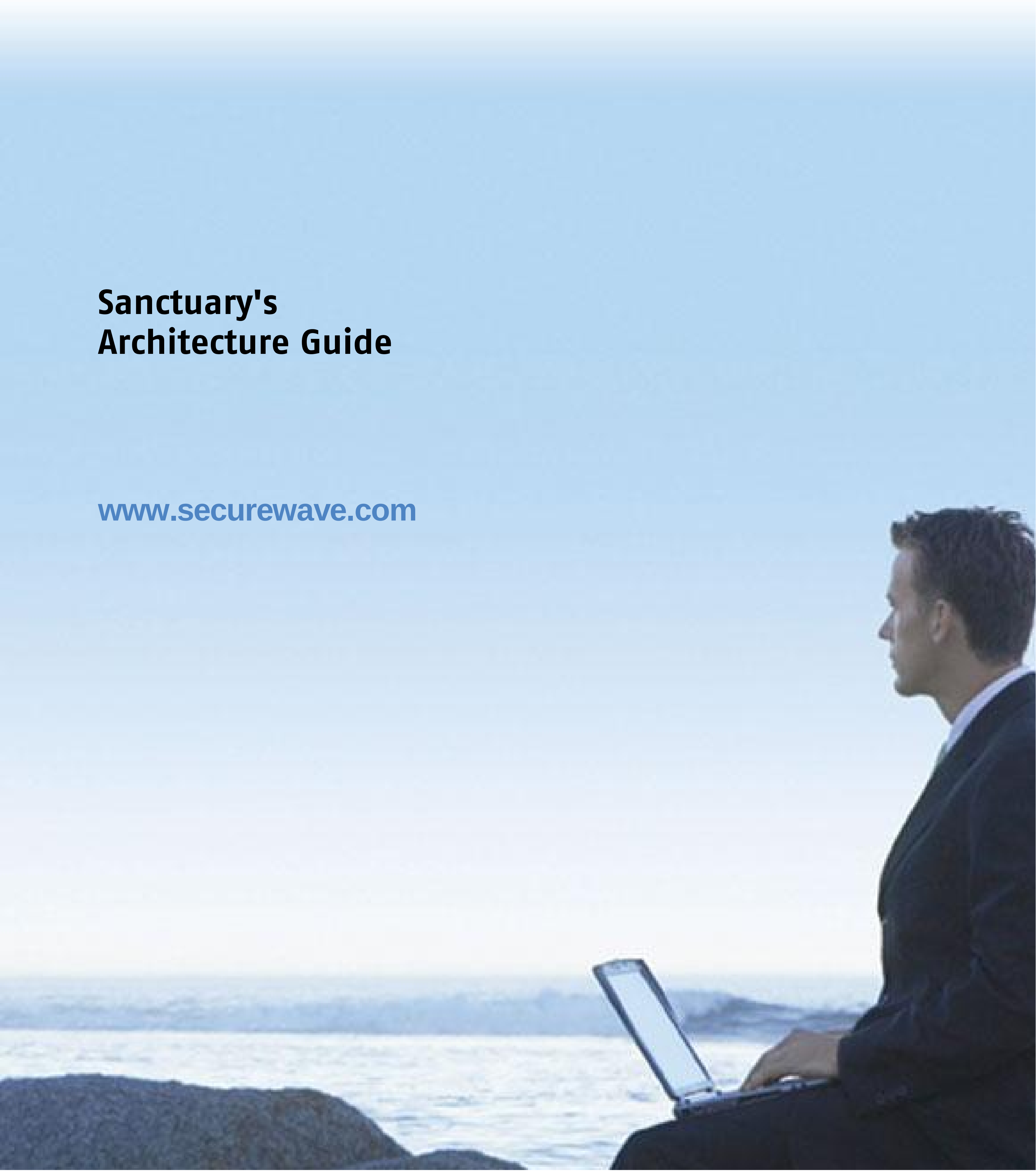
SecureWave

Safeguarding Tomorrow



Sanctuary's Architecture Guide

www.securewave.com

Liability Notice

Information in this manual may change without notice and does not represent a commitment on the part of SecureWave.

SecureWave, S.A. provides the software described in this manual under a license agreement. The software may only be used in accordance with the terms of the contract.

No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of SecureWave.

SecureWave claims copyright in this program and documentation as an unpublished work, revisions of which were first licensed on the date indicated in the foregoing notice. Claim of copyright does not imply waiver of other rights by SecureWave.

Trademarks

Sanctuary is a trademark of SecureWave, S.A.
All other trademarks recognized.

SecureWave, S.A.
Atrium Business Park
23, rue du Puits Romain
L-8070 Bertrange
Luxembourg

Phone:	+352 265 364-11 (add prefix 011 when calling from USA or Canada)
Fax:	+352 265 364-12 (add prefix 011 when calling from USA or Canada)
Web:	www.securewave.com

Technical Support hours are Monday to Friday, 8:00 to 20:00 CET/CEST in Europe and 8:00 AM to 8:00 PM ET/EDT in North America.

You can contact our technical support team by calling:

+352 265 364 300 (International),
+1-877-713-8600 (US Toll Free),
+44-800-012-1869 (UK Toll Free)

or by sending an email to support@securewave.com

Published on: August 2007

Contents

Introducing Sanctuary

The real world can be harsh: Trojans, worms, viruses, hackers, and even careless or disgruntled employees threaten your company's data and structure. They can undermine your business with extraordinary speed, and the cost and damage to applications, data, confidentiality, and public image, can be immense.

Your role, until now, has been to try to anticipate malicious code and actions before they occur and to react to them when they do — in a never-ending expenditure of time, money, and energy.

Sanctuary solutions stop that futile game for good. With Sanctuary software, you define what is allowed to execute on your organization's desktops and servers, and what devices are authorized to copy data. Everything else is denied by default. Only authorized programs and devices will run on your network, regardless of the source. Nothing else can get in. Nothing.

Sanctuary provides policy-based control for all devices and applications that can be used on enterprise endpoints. Using an automated whitelist approach, Sanctuary enables the development, enforcement, and auditing for application and device use in order to maintain IT security, reduce the effort and cost associated with supporting endpoint technologies, and ensure compliance with regulations. By using a whitelist approach, enterprises can literally turn their backs on the volumes of unwanted applications, malware, and unauthorized devices and instead focus on what is authorized and approved.

What makes Sanctuary so revolutionary is that it is proactive, not reactive. You are empowered, not encumbered. You lower and raise the drawbridge. You open and close the borders. You create calm in a chaotic world.

The whitelist approach

Concepts

A **black list** is a register of applications/devices that, for one reason or other, are being denied execution/access privileges.

White lists are the exact opposites of blacklists. Where a blacklist specifies which device/application is not allowed, while granting permissions to all others, a white list only allows access/execution rights to those who are already on the list, while denying permissions to all others.

A **grey list** is everything in between white and black lists. If application control cannot identify the application, then the user may place it on a grey list with extra auditing vigilance enabled so that IT can make a subsequent decision whether to authorize it or not.

Sanctuary works on the basis that the use of all executables and peripherals are denied unless explicitly authorized. An administrator initially creates — and then maintains, as needed — a **white List** of authorized executables/devices. This overcomes the time consuming administrative burden of constantly updating and maintaining a black list of executables that are not authorized to run.

Advantages/disadvantages of using a white list

The following table shows the advantages and disadvantages of the different approaches:

Blacklists (Reject list or block list. This allows everything that is not on the list.)		**White lists** (Accept list. This denies everything that is not on the white list.)	
Advantages	**Disadvantages**	**Advantages**	**Disadvantages**
Easy to install.	Exponential growth - consuming resources.	Can be created at computer, user, or user group (or specific device) level.	Takes longer to install and personalize.
	Updates are futile since there are always new, unknown, application and devices.	More secure.	
	Costly and complicated to maintain.	More accurate and granular.	
	Can only detect what is already known.	Inexpensive and simple to maintain.	
	Constant updates are required (but these do not block everything).	Easy to customize and manage.	
	Not ‘future-proof’.	Is only modified for specific cases.	
	Usually only ban applications/devices when it is too late!	Will not allow unknown application to execute or access to new devices not previously known.	
	You typically give the control to a third party that constantly updates this list.	Authorizing the use of a device/application is much easier than banning all those not accepted.	
	Cannot respond to ‘day-zero’ attacks.	It is a ‘future-proof’ approach.	
		It is almost ‘maintenance-free’ since the list only needs to be modified when a new application/device authorization is needed.	
		You have complete control over what in included in the white list.	
		‘Day-zero’ attacks are no longer a threat since everything is unless otherwise specified.	
		No definition updates are required.	

Table 1: Whitelist vs blacklist approach

Whitelist and blacklist examples

The traditional approach to computer security is to design a program to block out undesirable applications. Let’s assume you write such a program that is responsible for determining applications run or not. To maintain control you must provide a daily list of applications that are not allowed to run. When a user tries to run an executable, your software searches for it on the list and if it is there, prevent it from running. If a valid program is contaminated, your program cannot detect it since its name is not on the undesirable list (black list). It can run and create havoc in your network. Additionally, just because the program is not in the list does not mean that it is not a threat. You spend your weekends identifying these programs and constantly updating your list.

Let’s consider that you now try a different approach. You set a more flexible and general set of rules to determine what is allowed to run or not. Instead of only basing your assumption on a list, you instruct the program to also block all programs that behave strangely, have non-standard or suspicious names based on your experience-driven knowledge of computer security, and/or are on other black lists you can get your hands on. Your program now blindly blocks ‘almost’ all undesirable software but it also blocks some good ones in the process. Back to the drawing board to add even more rules and exceptions to your black list definition.

You now try a third tactic: You create a list of programs that *can* run (a white list), everything else is banned. You now use your weekends for your hobbies. Unless you explicitly modify your list, new threats pose no problem to your blocking software.

Since Sanctuary is based on a whitelist approach, you can configure it to authorize all acceptable applications/devices instead of blocking all those not tolerated.

A complete portfolio of security solutions

While our application control series steps-in whenever a user launch an executable to issue an "approved" or "unapproved" stamp, device control focuses on all those external removable devices that can be used as an open door from where data could escape or malicious code can enter.

Application control is a well-suited approach for those organizations that are looking for automated tools to help exercise tighter management execution control in their endpoints. On the other hand, device control goes from a simple device use blocking application to a full-blown device control application including encryption, auditing, logs, file filtering, shadowed data (a full copy of all data that enters/leaves premises), etc. Sanctuary combines the best of both worlds in a centrally administrated solution that can be used conjointly or each one of them as a separate solution:

	Product	**Target**
Sanctuary Application Control Suite	Sanctuary Application Control Server Edition	Prevents/denies unwanted executables within server environments, stopping attacks on mail servers, CRM applications, web and other critical database servers (Windows 2000 and Windows 2003).
	Sanctuary Application Control Terminal Services Edition	Extends the power of Sanctuary to the complex thin client terminal environment (both Windows® and Citrix®) by providing granular application and access control over users on business critical terminal services, enhancing availability and stability.
	Sanctuary Application Control Custom Edition	Enforces a granular policy over user/user groups and their use of specific applications to match complex and distributed enterprise environments (Windows 2000, Windows XP, Windows Vista, Windows XP Embedded, Windows 2003 and Novell workstations)
	Sanctuary Device Control	Seals security breaches by providing a complete USB security, port protection, and control of all removable devices across your network
	Sanctuary for Embedded Devices	Moves beyond the traditional desktop and laptop endpoints and onto a variety of platforms that include ATMs, industrial robotics, thin clients, set-top boxes, network area storage devices and the myriad of other systems running Windows XP Embedded

Table 2: A complete solution for all your needs

Each component is explained in the next sections.

Sanctuary Application Control Suite

Sanctuary Application Control Suite is an Application Execution Management solution that provides organizations with the capability of exercising total control over which applications can run on Microsoft and Novell based networks. Sanctuary Application Control Suite works on the basis that the use of all executables, scripts and macros is denied unless explicitly authorized. A white list of authorized files is created and maintained. This overcomes the time consuming administrative burden of constantly updating and maintaining a black list of executables, scripts and macros that are not authorized to run. Sanctuary

Application Control Suite also protects against tampering by using file integrity checking to ensure that authorized executables cannot be tinkered with.

Sanctuary Device Control

Sanctuary Device Control is a software component that extends the control of I/O devices policies. Based on a positive model, device access for users is prohibited by default. Only explicitly authorized devices can be accessed. Sanctuary Device Control manages access to devices by applying an Access Control List (ACL) to each device type. To grant access, the Administrator only needs to associate Novell objects (organizational units, users, user groups) with the devices and/or device classes which they are allowed to access.

Sanctuary for Embedded Devices

Sanctuary for Embedded Devices establishes a trusted device and applications environment based on Microsoft Windows Embedded platforms and never worry about the risk of data loss or malicious attacks that could cost your organization thousands of dollars in damages. Easily control your organization's entire thin client desktop configuration from one central location. Sanctuary for Embedded Devices offers you endpoint Security and Policy Enforcement for ATMs, KIOSKS, POS, Terminals, etc.

About this Guide

This guide provides an overview of the architecture of the Sanctuary solution - Sanctuary Application Control Suite (Sanctuary Application Control Server Edition, Sanctuary Application Control Custom Edition, and Sanctuary Application Control Terminal Services Edition), Sanctuary Device Control, and Sanctuary for Embedded Devices. Without this basic information, it is difficult to grasp just how powerful Sanctuary is.

- *Chapter 1: Sanctuary components* provides a high-level overview of the Sanctuary solution, how it works and the benefits it can provide to your organization.
- *Chapter 2: How Sanctuary works* explains the architecture and functionality of each Sanctuary component.
- The *Glossary* and indexes (*Index of Figures*, *Index of Tables*, and *Index*) provide quick access to specific terms or topics.

Typefaces

We use the following typefaces to differentiate between certain types of contents throughout this guide:

Italic	Represent fields, menu options, and cross-references
`Fixed width`	Shows messages or commands that should be typed at the command prompt
SMALL CAPS	Represents buttons you select

Symbol explanation

We use the following symbols to emphasize important points about the information you are reading throughout this guide:

✍	*Special note. This symbol identifies additional information about the topic reading. These notes may also relate to other parts of the system or be points that need particular attention.*
⌛	*Time saver. This symbol identifies 'short-cuts' or tips that may save you time.*
💣	*Caution. This symbol identifies potential risks when working with certain aspects of Sanctuary, for example where data may be lost or problems with the operation of your system may occur.*

Keyboard conventions

A plus sign between two keyboard keys means that you must press those keys at the same time. For example, ALT+R means that you hold down the ALT key while you press R.

A comma between two or more keys means you must press each of them consecutively. For example ‘Alt, R, U’ means that you press each key in sequence.

For more information

In addition to the documents and online help that come with Sanctuary, further information is available on our web site at: www.securewave.com

This regularly updated web site contains:

> The latest software upgrades and patches (for registered users).

> Troubleshooting tips and answers to frequently asked questions.

> Other general support material that you may find useful.

> New information about Sanctuary.

> Our Knowledge Base (KB), with FAQ (Frequent Asked Questions) and practical information of your everyday use of Sanctuary solutions.

To contact us

If you still have a question after reviewing the online help, documentation, or SecureWave knowledge base, you can contact your SecureWave customer support team by telephone, fax, email, or regular mail.

Technical Support hours are Monday to Friday, 8:00 to 20:00 CET/CEST in Europe and 8:00 AM to 8:00 PM ET/EDT in North America.

You can contact our technical support team by calling:

+352 265 364 300 (International),
+1-877-713-8600 (US Toll Free),
+44-800-012-1869 (UK Toll Free)

or by sending an email to: support@securewave.com

Alternatively, you can write to customer support at:

SecureWave, S.A.
Atrium Business Park
23, rue du Puits Romain
L-8070 Bertrange
Luxembourg

Chapter 1: Sanctuary components

This chapter introduces the powerful Sanctuary solution and explains its infrastructure in detail. A Sanctuary solution includes the following four main components:

- One *SecureWave Sanctuary Database*. This holds device and/or executable authorization information.
- One or more *SecureWave Application Servers* with one or more *Data File Directories* (DFDs). These act as an intermediate between the Sanctuary Client Driver (see below) and the SecureWave Sanctuary Database. It distributes the list of devices and/or software permissions for each client computer and/or User/User Group.
- The *Sanctuary Client Driver*: This enforces the centrally-defined policies on the machines you want to protect from using unauthorized software/devices. The client communicates with the SecureWave Application Server to get the list of authorized software/devices.
- Administrative tools, in particular the *Sanctuary Management Console*. This centrally configures Sanctuary policies, and manages the day-to-day administrative tasks and procedures of policy enforcement.

The following diagram shows a typical Sanctuary infrastructure. Each implementation may have more than one SecureWave Application Server and a SecureWave Sanctuary Database connected over a wide area, therefore making Sanctuary software very scalable:

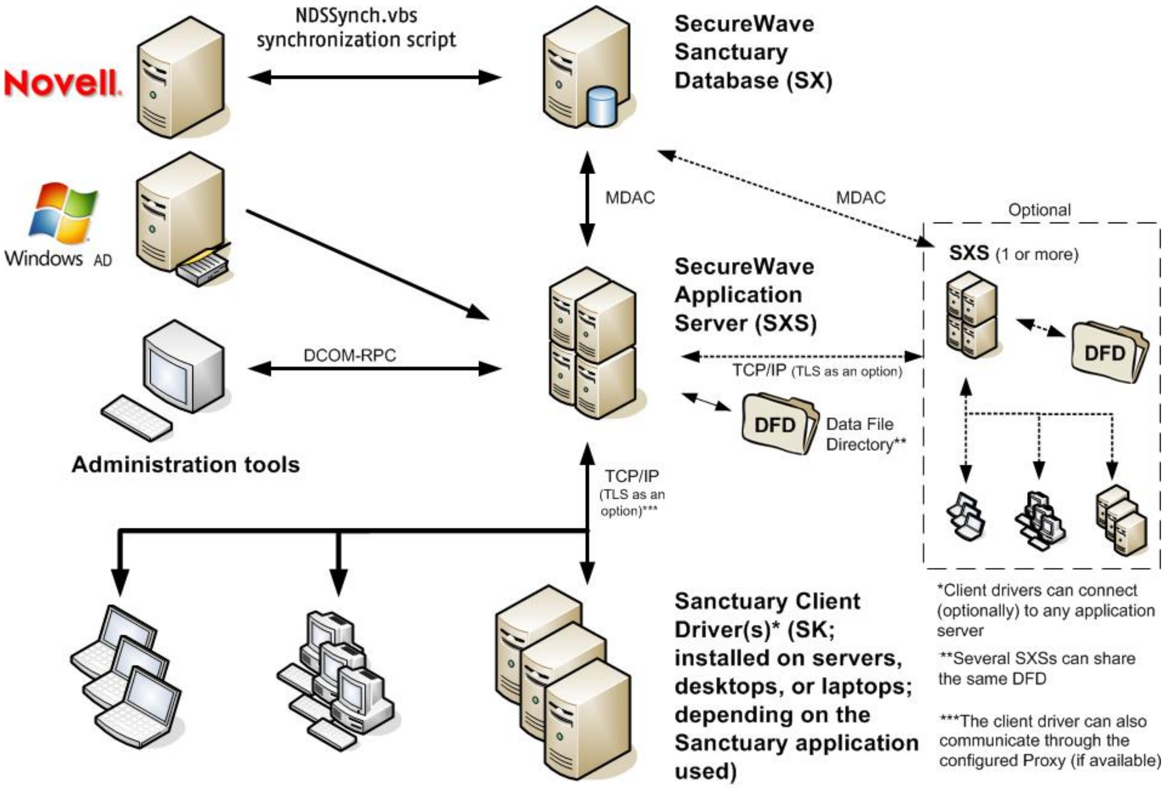

Novell.
NDSSynch.vbs synchronization script
SecureWave Sanctuary Database (SX)
Windows AD
MDAC
MDAC
Optional
SXS (1 or more)
SecureWave Application Server (SXS)
DFD
DCOM-RPC
TCP/IP (TLS as an option)
DFD
Data File Directory**
Administration tools
TCP/IP (TLS as an option)***
Sanctuary Client Driver(s)* (SK; installed on servers, desktops, or laptops; depending on the Sanctuary application used)
*Client drivers can connect (optionally) to any application server
**Several SXSs can share the same DFD
***The client driver can also communicate through the configured Proxy (if available)

Figure 1: Sanctuary components

We explain each of these components in the following sections.

The SecureWave Sanctuary Database

The **SecureWave Sanctuary Database** serves as the central repository of authorization information, such as lists of executable files, scripts and macros, the digital signatures ('hashes') that uniquely identify these files, File Groups, authorized users and User Groups, device permissions, and user policies. It also stores audit logs of administrator's actions.

This database is built on the Microsoft SQL Server 2000/2005, 2005 Express Edition, or Microsoft Database Engine (MSDE) 2000. For organizations with fewer than 200 users, the MSDE 2000 or SQL Server 2005 Express Edition is sufficient. Larger organizations must use Microsoft SQL Server.

The SecureWave Application Server

Each Sanctuary installation requires at least one SecureWave Application Server and related *Data File Directory* (which may or may not be on the same machine) to store log information. All servers can either write to the **same** shared directory, or alternatively, a different one for each server (see *Figure 1*). The **SecureWave Application Server** communicates between the SecureWave Sanctuary Database and the protected servers or computers.

The SecureWave Application Server component runs as a Windows Service under any domain user account capable of reading Domain users/groups/computers accounts from the Domain Controller. It performs the following functions:

> Gets the latest information about access privileges and device I/O permissions from the database and stores it in its cache.

> Signs or encrypts the list, compresses it, and passes the updated access information list to servers and computers, where it is also stored locally. (The updates contain the changes to the permissions rather than the whole list.)

> Saves a log of administrators and, optionally, users actions (including information about where application or device access have been denied).

The SecureWave Application Server runs as a service and keeps track of the connected clients and their status, coordinating data flow between SecureWave Application Servers — if you are using more than one — and the SQL database. As with other TCP-based services, SecureWave Application Server cannot handle clients connecting through a firewall or proxy unless the required ports are opened. By default, it uses port 65129 or 65229 (for the TLS protocol) to listen to clients' or other SecureWave Application Server's requests. Clients use port 33115, by default, to receive information (and respond if it is the SecureWave Application Server who initiated the communication). These three ports are required for a full two-way communication. You can configure these ports to suit your needs (see Sanctuary's Setup Guide for more information).

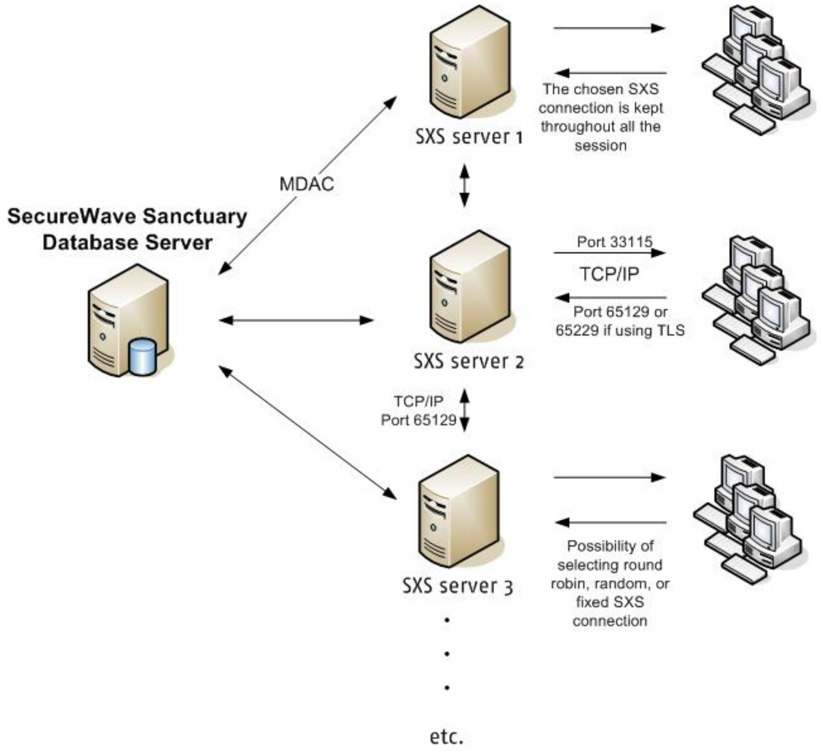

SXS server 1
The chosen SXS connection is kept throughout all the session
MDAC
SecureWave Sanctuary Database Server
Port 33115
TCP/IP
Port 65129 or 65229 if using TLS
SXS server 2
TCP/IP
Port 65129
SXS server 3
Possibility of selecting round robin, random, or fixed SXS connection
etc.

Figure 2: SecureWave Application Server-client-Sanctuary Client Driver X intercommunication

Up to three different SecureWave Application Servers can be defined in the setup of a client — using fixed IP addresses or DNS aliases. Additional servers can be assigned either by changing an option in the management console or via a registry key. If no SecureWave Application Server is available at logon, the client driver falls back on the permission list that was stored on disk during the last successful connection. If no such list exists, the client driver institutes a complete lockdown of all devices/applications. Permissions lists can be imported into a computer if required, for example, when no server is available because the machine is disconnected from the network.

DNS is only indirectly used to look up an IP addresses for a computer that must be accessed. If the corresponding entry in the server list is a DNS name, it is resolved, and the first returned IP address is chosen, as required by round-robin DNS conventions. A connection is then attempted. If it works, execution then proceeds normally.

If the connection fails, the client selects the next server from its list and repeats the process. If the end of the list is reached, the client uses the local permission list, as previously explained. This behavior is controlled by the *FirstServer* registry key.

This server also receives client’s logs and shadow information — in compressed format — that is safely stored in a common data file directory (DFD) defined at setup time — also in compressed format.

How permissions are defined, managed, and stored

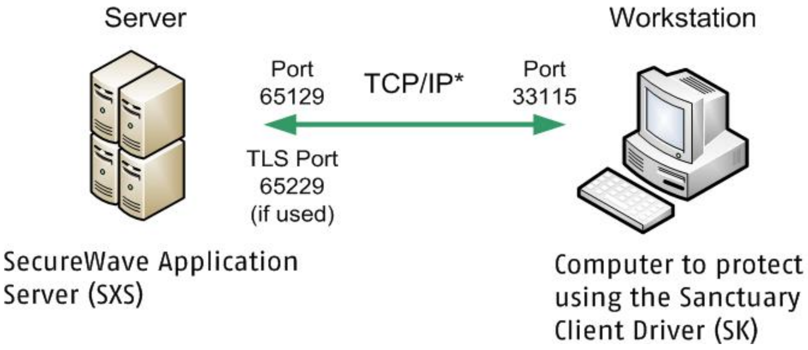
Once the Sanctuary Client Driver installed, it manifests itself as an icon on the system tray. This informs the user about permission changes, by means of pop-up messages.

Using the limited user interface, users cannot change permissions, they can only ask for available updates that have been defined by an administrator. When using any of the components of Sanctuary Application Control Suite, the user also has the option, if the administrator decides to grant this privilege, of accepting or denying execution of applications, scripts and macros. If all decisions are left to the user’s discretion, they can control them using the client’s available options. Nothing else is allowed.

To change permissions, a Sanctuary administrator uses a management console to interact with SecureWave Application Server that, in turn, communicates with the database and the clients.

Permission changes are sent to users at the next event — for example, when the user logs in — or, as an alternative, the administrator can ‘push’ them to all computers, specific ones, or export them for a later importation on the client(s). The SecureWave Application Server informs all online clients when new permissions become available, or sends them if specifically asked by the user — a ‘push-pull’ mechanism.

The following diagram shows a view of the client/application server relation:


Server
Workstation
Port 65129
TCP/IP*
Port 33115
TLS Port 65229 (if used)
SecureWave Application Server (SXS)
Computer to protect using the Sanctuary Client Driver (SK)

*If the SXS server initiates the communication, it uses port 33115 and expects that the client responds using the same port. If the client driver initiates the communication, it uses port 65129 or 65229 (if TLS is used).

The communication can also be done using the proxy server configured for IE (only through TLS, port 443).

Figure 3: Sanctuary Client Driver/SecureWave Application Server relation

Sanctuary Client Driver

The **Sanctuary Client Driver** is installed on each server & computer you want to protect. This client component runs as a kernel driver on Windows XP/2000/2003/Vista:

If you are using Sanctuary Application Control Suite, the Sanctuary Client Driver does the following:

- Calculates the digital signature (‘hash’) of files loaded for execution.
- Checks that hash against the locally stored authorization list (of hashes for executables, scripts of application files in which VBA macros are embedded).
- Ensures that only authorized executable files can run.
- Bans and logs any attempts to run unauthorized files.
- Optionally, permits local authorization of a denied file.
- Generates log records of all application access attempts — approved and denied. The *Log Access Denied* option is enable by default.

If you are using Sanctuary Device Control, the Sanctuary Client Driver:

- Ensures that only those I/O devices that the user has been authorized to use can be accessed on the computer. Any attempt to access an unauthorized device is barred, regardless of the computer the user logs on to.

The communication component of the client, SCC, which runs as a service, sends log data that can be viewed via the management console.

End-users cannot interact with the Sanctuary Client Driver, except to receive notifications when their permissions changes or to update them using the *Refresh Settings* command of the system tray icon. The user cannot change in any way its settings or permissions.

The client is installed on each computer you want to control. The setup also installs an application that provides (optionally) device status information to the end-user.

The administrator can also ask the user to show the ‘salt‘ value used to do endpoint maintenance when the computer is not connected to the network and this value cannot be obtained by alternative methods. See the administrator’s guides for more info.

The client function (device and/or application blocking) is defined by the product license. Depending upon the licensed Sanctuary components, the client blocks devices/media and/or applications. The client is divided into three primary components (see *Installed components* on page *13*):

1. Kernel driver (sk) — enforces defined Sanctuary policies.

2. Communication service (scomc) — provides communication with the SecureWave Application Server(s).

3. User interface (RtNotify) — provides status information and notifications to the user.

The key is that even if the communication service or user interface is disabled, the kernel driver is still protecting the managed device. For example, should a user manage to disable the user interface, protection remains in force and the 'least privilege principle' — denying anything not expressly permitted — is applied. This means that components are protected against tampering by users (using Sanctuary's client hardening functionality).

The following diagram shows this layered relationship:

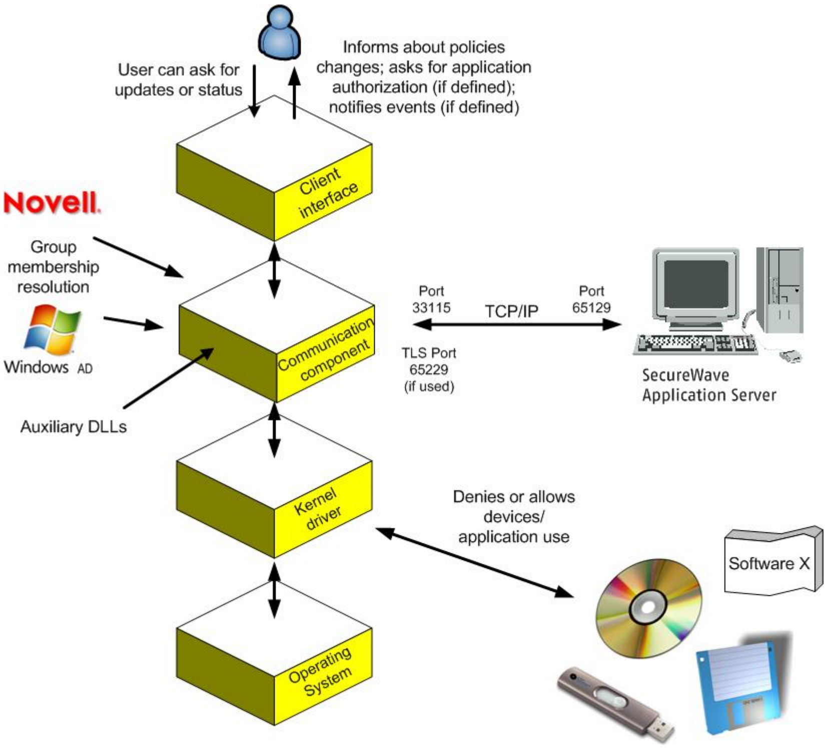

User can ask for updates or status
Informs about policies changes; asks for application authorization (if defined); notifies events (if defined)
Client interface
Novell.
Group membership resolution
Windows AD
Auxiliary DLLs
Communication component
Port 33115
TCP/IP
Port 65129
TLS Port 65229 (if used)
SecureWave Application Server
Kernel driver
Denies or allows devices/ application use
Software X
Operating System

Figure 4: Client layered model

Installed components

The following client components are installed on a Sanctuary-protected computer:

> RtNotify.exe — This is the primary user interface that informs the user of updated policy changes completed by the administrator (these messages can be deactivated). It displays itself as an icon (that can optionally be disabled) in the Windows system tray.

> Sk.sys — This is the kernel component that is responsible for enforcing the centrally defined policies, by determining which applications and/or devices can be accessed. It has no user interface.

> Scomc — This component is responsible for communication with the SecureWave Application Server(s). It has no user interface.

> Auxiliary DLLs — These provide features additional to the 3 core components defined above. The files contain support for RtNotify localization information, 16-bit application control, and macro and script protection. They have no user interface.

Protocol and ports

Sanctuary is based on standard TCP/IP protocols for all communication between clients and servers. TCP/IP was chosen due to its pervasive implementation throughout most IT infrastructures. Currently Sanctuary uses only two configurable ports for full two-way communication between the client and server components.

Internet protocols were first developed in the mid-1970s. They are now the most widely used open-system (nonproprietary) protocols since there are equally well suited for LAN or WAN communication. Internet Protocol (IP, layer 3 of the OSI model) contains addressing and control information and forms the heart of the Internet protocols, along with the Transmission Control Protocol (TCP, level 4 of the OSI model).

Using the TCP/IP protocol offers some clear advantages over other protocols, including the following:

- It allows enterprise networking connectivity between Windows and non-Windows based computers.
- It can be used to create client/server applications.
- It is reliable.
- It is easily expandable.
- It has good failure recovery.
- It has a high error-rate handling… and so on.

When installing the Sanctuary Client Driver on your protected machines, TCP/IP should already be activated and configured. Since almost all modern networks use these protocols, this should already be the standard setting in your network.

As an alternative, to reinforce security levels, you can select the TLS communication protocol. This means that all communication between clients and the SecureWave Application Server is encrypted rather than communications only being signed before transmitting. A Certificate Authority must emit a certificate if you plan to use TLS.

SecureWave Application Server incorporates a high-performance built-in TCP server. It uses this to maximize throughput for client driver requests. This TCP component can be fine-tuned to accommodate nearly all possible configurations.

Sanctuary’s client, by default, uses port 33115 to listen to the SecureWave Application Server while this component uses port 65129 (or 65229 if you are using the TLS protocol) to communicate with the Sanctuary Client Driver. When installing the client on a Windows XP SP2 (or Windows 2003 SP1 with the firewall enabled) it is important to open these ports otherwise the client will be blocked with the most restrictive policies (those defined when installing it) or the last permission list locally stored. See *Sanctuary's Setup Guide* for more information on how to open these ports.

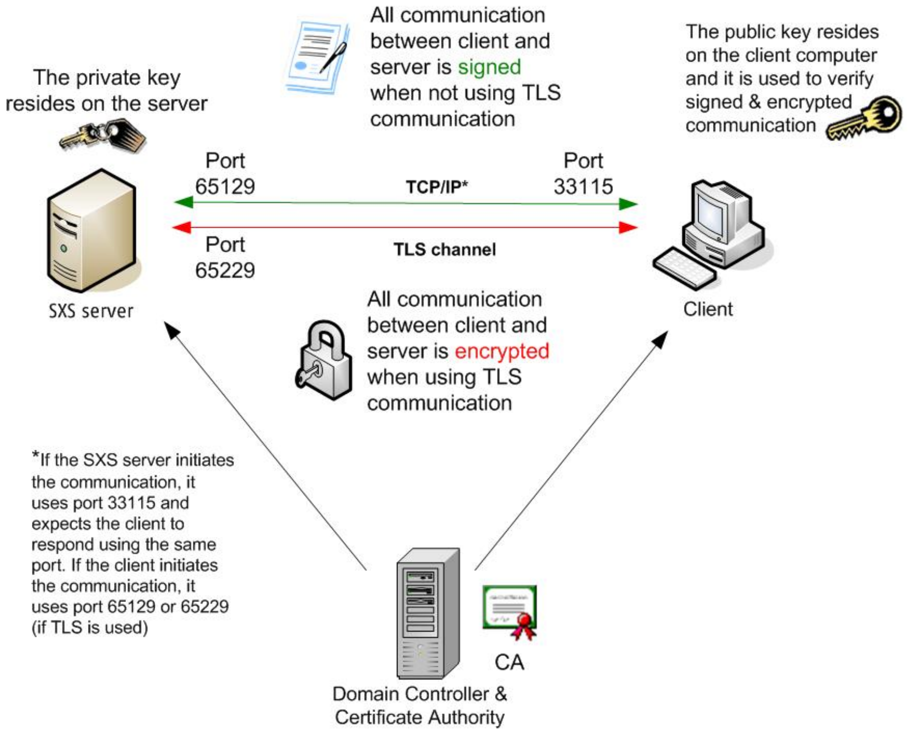

The private key resides on the server
All communication between client and server is signed when not using TLS communication
The public key resides on the client computer and it is used to verify signed & encrypted communication
Port 65129
TCP/IP*
Port 33115
Port 65229
TLS channel
SXS server
Client
All communication between client and server is encrypted when using TLS communication
*If the SXS server initiates the communication, it uses port 33115 and expects the client to respond using the same port. If the client initiates the communication, it uses port 65129 or 65229 (if TLS is used)
CA
Domain Controller & Certificate Authority

Figure 5: TLS and non-TLS communication between the Sanctuary Client Driver and the SecureWave Application Server

Operation overview

As a user logs into the computer, several background actions are carried out before the operating system completely boots and can access the installed programs/devices. These are as follows:

1. The system checks that all client components are present and refuses to load the operating system if the one of them is missing or has been tampered with.
2. The client checks that a valid SecureWave Application Server exists and is reachable over the network. If unavailable, the client uses the previously cached internal permissions list. If a SecureWave Application Server can be reached, the client identifies itself and requests a permission list update.
3. If the client does not have the latest permissions, it requests an update. The SecureWave Application Server reacts to this by retrieving the list from the database (only if its cache is empty or has been modified). The SecureWave Sanctuary Database returns the requested list.
4. The SecureWave Application Server stocks the new permission list in its cache, selects what has changed, compresses it, signs or encrypts the resulting list (depending if you installed the clients with TLS communication or not) and then sends it to the client.
5. The client replaces its current permission list with the new one.
6. If the user logs off, the client informs the SecureWave Application Server.

The client sends activity logs (by request and subject to certain options defined by the administrator) to the SecureWave Application Server. The client is also responsible for saving, parsing, compressing, and sending shadow (a copy of transferred data to devices) and log information to the SecureWave Application Server.

Since the Sanctuary Client Driver is the first one to be loaded, there is no potential risk of the user trying to intercept or deactivate it. To protect it further, a administrator can choose to select a ‘client hardening’ policy where even users with administrator’s rights cannot uninstall the client without a prior permissions ‘ticket’. See Sanctuary Application Control Suite Administrator's Guide and Sanctuary Device Control Administrator's Guide for more information about ‘hardening’ the client and sending ‘endpoint maintenance tickets’.

Key usage

As the SecureWave Application Server is the one that sends all permissions/rules to the client, it is important to secure this communication. This is done by means of a public/private encryption key pair generated using the Rivest-Shamir-Adelman (RSA) algorithm with a key size of 2,048 bits. This key pair, usually generated during the Sanctuary installation process, is used to assure the integrity of the communication between the SecureWave Application Server and its clients. The key pair is also used to encrypt media when using Sanctuary Device Control.

When starting SecureWave Application Server, it will check for the key pair in the following locations:

1. In the directory where the SecureWave Application Server executable is (usually %SYSTEMROOT%\SYSTEM32).
2. In the SecureWave Application Server’s private directory (%SYSTEMROOT%\SXSDATA).
3. In all removable drives and DVDs/CDs, in alphabetical order.

The search stops when the first valid key pair is found. If a higher level of protection is required, we strongly recommend storing the server’s private key externally to the SecureWave Application Server — for example on a CD, USB key, or floppy disk. Only the public key is available to the clients. The private key should only be available to the SecureWave Application Servers, either internally or externally.

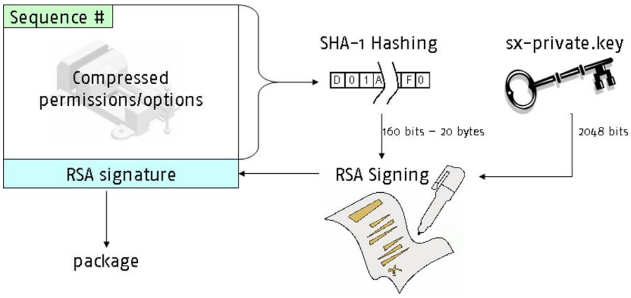


Figure 6: Building the client package

If the SecureWave Application Server is not reachable

When the client tries to communicate with the SecureWave Application Server, it does so by using the Fully Qualified Domain Name (FQDN) address(es) configured during the client setup (IP addresses do not work if you are using TLS protocol, see the Sanctuary's Setup Guide for further details.)

The FQDN addresses may or may not be active when the Sanctuary Client Driver tries to establish the communication. They may not be active, particularly when using remote clients through a Virtual Private Network (VPN) connection that does not have a physical cable connecting the server(s) to the client's machine or a firewall is blocking the required ports and they should not be opened for security reasons. In these cases, all communication is done using the Internet and, possibly, a proxy that acts as a barrier between the internal network and Internet since many corporations use proxy servers to manage various communication protocols and add a higher level of security to their network environment.

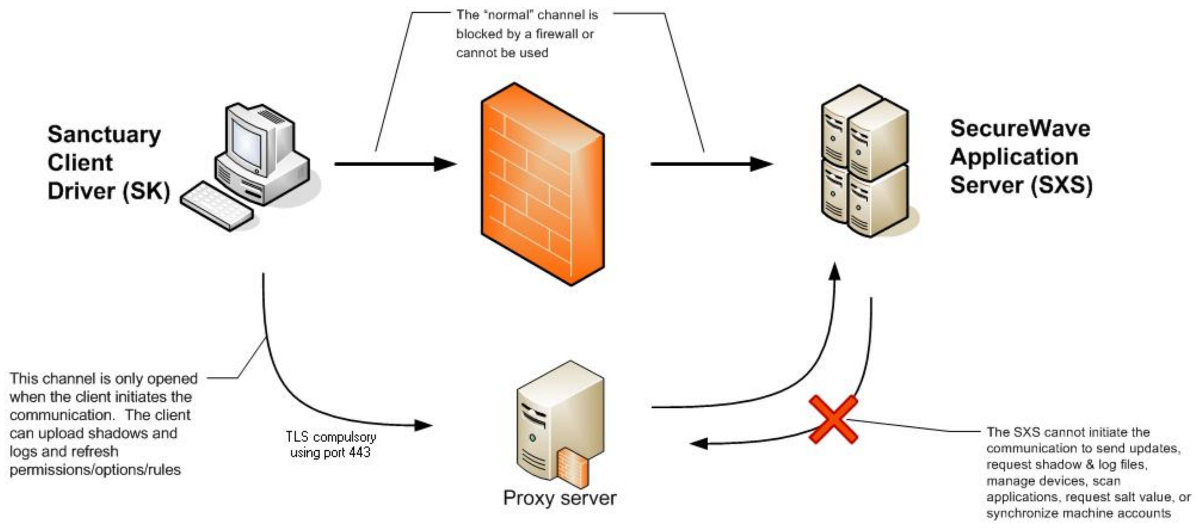

The "normal" channel is blocked by a firewall or cannot be used
Sanctuary Client Driver (SK)
SecureWave Application Server (SXS)
This channel is only opened when the client initiates the communication. The client can upload shadows and logs and refresh permissions/options/rules
TLS compulsory using port 443
Proxy server
The SXS cannot initiate the communication to send updates, request shadow & log files, manage devices, scan applications, request salt value, or synchronize machine accounts

Figure 7: Proxy use

Using a proxy to establish client-SecureWave Application Server communications

If the Sanctuary Client Driver cannot establish communication using the defined FQDN, it tries to use the proxy configured for the Internet Explorer to reach the SecureWave Application Server address(es) as shown in *Figure 7*. If this also fails, the SecureWave Application Server is considered unreachable and cached local policies apply to control application/device use.

> ✍ *If you are using Sanctuary Device Control and you defined offline or online device permissions, they will be enabled (depending of the 'Online/Offline State Detection' option configuration). See the Sanctuary Device Control Administrator's Guide for details.*

The use of a proxy is only valid when the Sanctuary Client Driver initiates the communication process — the user asks for a permissions refresh using the Sanctuary's tray bar icon — and not the other way around. The client driver can upload shadow and log files as well as refresh permissions/options/rules. The downside of using a proxy is that Sanctuary administrators cannot initiate a communication to request shadow and log files, manage devices (except using the Log Explorer module of the Sanctuary Management Console), send updates, scan applications to authorize them (when using Sanctuary Application Control Suite), retrieve salt value for client hardening disabling, or synchronize machine accounts.

If you want to take advantage of using a proxy, you must install Sanctuary Client Driver in TLS mode and configure the SecureWave Application Server's TLSPort registry key to 443 (see Sanctuary's Setup Guide). This port is used for secure web browser communications and should be configured for the Sanctuary Client Driver, SecureWave Application Server, and proxy. To be able to use this port, a valid machine's certificate must be present — which is already the case when installing SecureWave Application Server and Sanctuary Client Driver in TLS mode. Data transferred across such connections are highly resistant to eavesdropping and interception. Moreover, the identity of the remotely connected server can be verified with significant confidence. Web servers offering to accept and establish secure connections listen on this port for connections from web browsers desiring strong communication security.

When using the proxy, the client mimics Microsoft Internet Explorer proxy configuration. To configure, open Microsoft's Internet Explorer and then select *Tools* ➔ *Internet Options* ➔ *Connections* ➔ *LAN settings*. This behavior can be done in three distinctive modes — please refer to *Figure 11*:

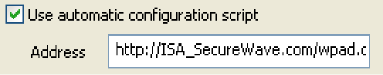
> Automatic mode (*Automatically detect settings*) – proxy configuration is done using a DHCP (Dynamic Host Configuration Protocol) server — follow the steps outlined in section *Configuring your DHCP server and proxy* on page 18.

☑ Automatically detect settings

Figure 8: Proxy configuration: Automatic mode

> Using the automatic configuration script (*Use automatic configuration script* — Web Proxy Automatic Discovery values, WPAD) — follow the steps outlined in section *Configuring your DHCP server and*

proxy on page *18* and then fill the address field with http://name_of_your_proxy/wpad.dat. In our example, this is the address of the ISA proxy ‘ ISA_SecureWave’ http://ISA_SecureWave.com/wpad.dat.

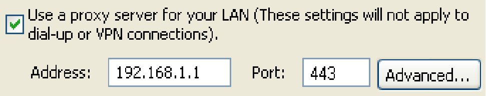


☑ Use automatic configuration script

Address: http://ISA_SecureWave.com/wpad.c

Figure 9: Proxy configuration: Automatic mode

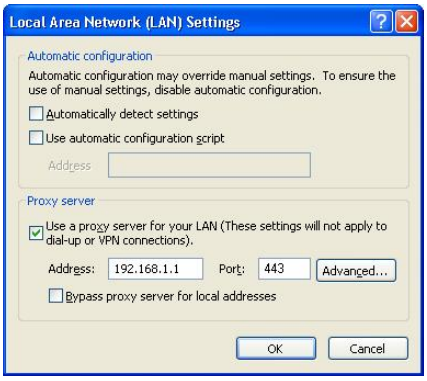
> Manual configuration mode (*Use a proxy server for your LAN*) using Secure HTTPS address — type-in the proxy address: the only one that is going to be used will be the secure one (you can check them by clicking the ADVANCED button), all others are not used for Sanctuary (HTTP, FTP, Gopher, and Socks).



☑ Use a proxy server for your LAN (These settings will not apply to dial-up or VPN connections).

Address: 192.168.1.1 Port: 443

Figure 10: Proxy configuration: Automatic mode



Local Area Network (LAN) Settings

Automatic configuration

Automatic configuration may override manual settings. To ensure the use of manual settings, disable automatic configuration.

□ Automatically detect settings

□ Use automatic configuration script

Address:

Proxy server

☑ Use a proxy server for your LAN (These settings will not apply to dial-up or VPN connections).

Address: 192.168.1.1 Port: 443

□ Bypass proxy server for local addresses

Figure 11: Proxy configuration (from Microsoft’s IE)

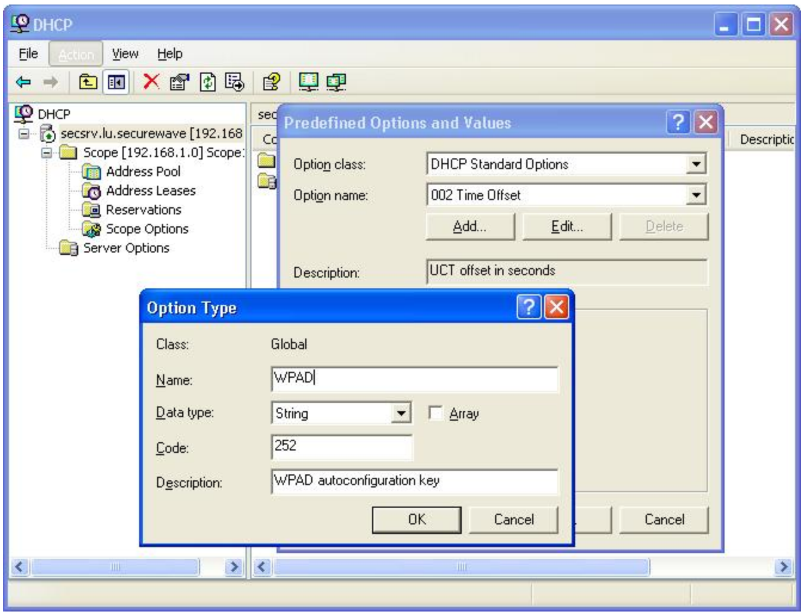
You must have Microsoft Internet Explorer version 6, or 7 for a proxy connection to work and the Sanctuary Client Driver should be installed on a Windows 2000 (SP4 or later) or Windows XP (SP1 or later) operating system.

Configuring your DHCP server and proxy

If you decide you want to use the proxy communication option, you must first configure your DHCP server and proxy. The manipulations are straightforward and simple:

1. Define a new Web Proxy Automatic Discovery option: in the DHCP console tree select the applicable DHCP server and then, on the *Action* menu, select *Set Predefined Options*.

2. In the *Predefined Options and Values* dialog click on ADD and complete the values as shown in the following image and close all dialogs by clicking on OK.



Option class: DHCP Standard Options

Option name: 002 Time Offset

Description: UCT offset in seconds

Class: Global

Name: WPAD

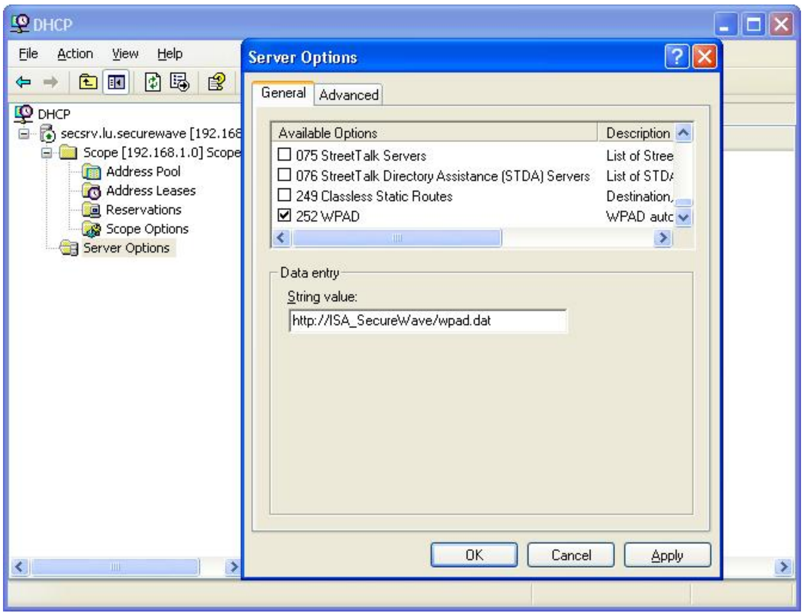
Data type: String □ Array

Code: 252

Description: WPAD autoconfiguration key

Figure 12: DHCP configuration: Define WPAD value

3. You must now activate the scope option. To do this, right click on the *Server Options* branch, select *Configure Options*, traverse the list until you find the WPAD value (the last one), and type your proxy address in the *String Value* field (this example uses a Microsoft Internet Security and Acceleration Server 2006 — ISA — proxy):

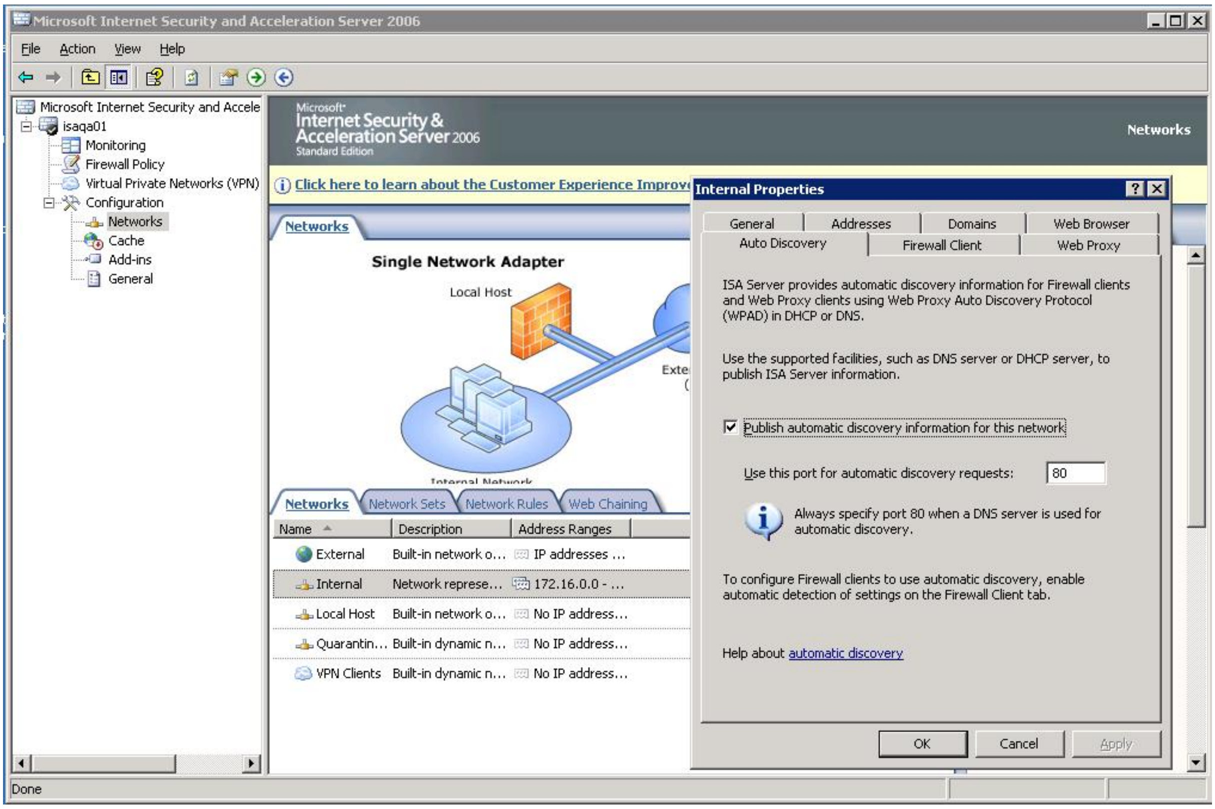


Available Options	Description
□ 075 StreetTalk Servers	List of Stree
□ 076 StreetTalk Directory Assistance (STDA) Servers	List of STDA
□ 249 Classless Static Routes	Destination,
☑ 252 WPAD	WPAD autc

String value: http://ISA_SecureWave/wpad.dat

Figure 13: DHCP configuration: Activate scope

4. Publish the ISA server information:



Name	Description	Address Ranges
External	Built-in network o...	IP addresses ...
Internal	Network represe...	172.16.0.0 - ...
Local Host	Built-in network o...	No IP address...
Quarantin...	Built-in dynamic n...	No IP address...
VPN Clients	Built-in dynamic n...	No IP address...

ISA Server provides automatic discovery information for Firewall clients and Web Proxy clients using Web Proxy Auto Discovery Protocol (WPAD) in DHCP or DNS.

Use the supported facilities, such as DNS server or DHCP server, to publish ISA Server information.

☑ Publish automatic discovery information for this network

Use this port for automatic discovery requests: 80

Always specify port 80 when a DNS server is used for automatic discovery.

To configure Firewall clients to use automatic discovery, enable automatic detection of settings on the Firewall Client tab.

Help about automatic discovery

Figure 14: ISA configuration: Publish server info

Sanctuary Management Console

The **Sanctuary Management Console** provides the administrative interface to the SecureWave Application Server. This tool — which can be installed on one or more computers — is used to configure the solution and perform a range of day-to-day administrative tasks. You can use it to:

> Define Administrator roles.

> Monitor system activity logs and option settings.

> Get standard reports or custom reports.

If you are using Sanctuary Device Control, you can also:

> Manage access to I/O devices.

> Authorize specific DVDs/CDs to be used in DVD/CD drives.

> Encrypt removable media.

> Grant users’ permission to use specific authorized DVDs/CDs or encrypted media.

> View lists of files transferred using authorized I/O.

> View the content of files transferred using authorized I/O.

> View information about attempts to access or connect to unauthorized devices.

If you are using Sanctuary Application Control Suite, you can also:

- Build lists of executable files, scripts and macros to be managed.
- Organize those authorized files into logical File Groups.
- Assign File Groups to users and User Groups.
- Manage and maintaining the authorization database.

The Sanctuary Management Console and SecureWave Application Server are linked through the RPC level 6 protocol (fully encrypted messages). The unique architecture of the Sanctuary solution generates minimal network traffic, so you do not need high-speed connections.

Each protected server and computer client maintains its own local authorization copy, so routine application requests do not have to traverse the network. Only log files and periodic differential updates are sent to them.

Administration Tools

When you install the Sanctuary Management Console, you also install other tools to manage the system. Tools that are common to all Sanctuary applications include the following (see *Sanctuary's Setup Guide* for more information):

- The *Client Deployment Tool*. This can be used to install the Sanctuary Client Driver on your protected computers and servers. It uses standard MSI technology. You can also use it to find out which computers already have the client installed and its status.
- The *Key Pair Generation*. This utility is used to create a unique set of private and public keys to assure communication between the SecureWave Application Server and the Sanctuary Client Driver.

  ✍ *You should always generate your own set of keys before deploying the product in a working environment.*

- The *SXDomain* command-line domain synchronization tool. This informs the SecureWave Sanctuary Database of changes made to the domains, users, groups, and workstations within your network.
- *Novell's Synchronization Script*: This is the command-line tool used to synchronize Novell's eDirectory objects (OU, group, user, and workstations) so that an administrator can manage them and deny/allow execution access to applications in a Novell environment.

If you are installing Sanctuary Application Control Suite, the following tools may also be installed (see the *Sanctuary Application Control Suite Administrator's Guide* for more information):

- The *Authorization Wizard*. The first step when authorizing a file to run is to identify its digital signature (hash) and compare this to a list of authorrized file hashes. You can use the Authorization Wizard to spot files copied to computers by installation routines, and incorporate their hashes to the SecureWave Sanctuary Database. The source can be either the original CD/DVD-ROM or the files held on a target system hard drive.
- The *Authorization Service Tool* is used to monitor changes and create updates (using Microsoft's SUS or WSUS).
- The *Versatile File Processor Tool* is used, either with the Authorization Service Tool or independently, to scan files.
- The *File Import/Export Tool* is used when updating from another Sanctuary system or to populate a SecureWave Sanctuary Database with already defined File Groups and hashes.

If you are installing Sanctuary Device Control, the following tool may also be installed (see the *Sanctuary Device Control Administrator's Guide* for more information):

- *Sanctuary Device Control Stand-Alone Decryption Tool* (SADEC). This can be used to decrypt removable devices in those organizations where Sanctuary is not installed. (The user needs administrative rights to install this tool. Alternatively the administrator can opt to use another encryption schema that does not need this tool, or administrative rights.)

Network communications

Sanctuary Client Driver communications

The Sanctuary Client Driver acts both as a client and a server. As a client, it contacts the SecureWave Application Server whenever there is a need (requesting hash-lists, devices Access Control Lists (ACL), reporting log-on and log-offs, uploading log files, etc.). As a server it awaits messages from the SecureWave Application Server that update part or all of its local store of hash-lists, ACLs and option settings.

Connections are created on a per-request basis since the time that the SecureWave Application Server and the Sanctuary Client Driver spend in communication is negligible compared to the time they do not generate network traffic.

SecureWave Application Server communications

The SecureWave Application Server, internally, consists of two distinct subsystems. One handles requests from administrative clients and exposes its services via a secure, authenticated Remote Procedure Call (RPC), the other one communicates with Client Drivers.

RPC server: In the SecureWave Application Server, authenticated RPC is used to expose administrative functionality, in particular the interfaces required to browse and manage the hashes and file groups in the database, and to offer control over driver behavior.

TCP Server: The SecureWave Application Server offers a TCP/IP server based on Microsoft Windows I/O Completion Ports (IOCP), the highest-performance thread and I/O management option that Microsoft Windows offers to applications. The most important server tasks are responding to log-on and log-off notification messages from Sanctuary Client Driver, i.e. processing start (boot) and stop (shutdown) messages from them, and creating and dispatching hash-lists at a client driver's request.

TCP Client: The TCP/IP client built into the SecureWave Application Server mainly serves to push updates to client drivers. When an administrator makes changes to options or permissions, client drivers may need to be notified of such changes immediately. For permission changes, this will typically also invalidate the hash-list cache mentioned before.

No broadcasting: Internally, the SecureWave Application Server uses a thread pool to perform mass updates. It connects to each Sanctuary Client Driver individually according to the driver's state (the database keeps track of drivers and users that are on-line). This is more work than broadcasting, but offers the advantage of guaranteed delivery, a feature not found in broadcast-capable protocols.

Inter-server communications: The forcing of updates mentioned above also raises a need for multiple instances of the SecureWave Application Server to communicate among themselves. In particular, when an administrator requests an immediate hash-list update, the instruction to flush the hash-list cache must be relayed to every server in order to keep the caches coherent. Since all servers share a common database, they all register themselves in that database. Intra-server notifications are sent through their TCP/IP channel.

Chapter 2: How Sanctuary works

This chapter contains a high-level summary of the behind-the-scenes workings of this powerful yet easy-to-use security solution.

Sanctuary Application Control Suite

Sanctuary is an operating system extension solution that enforces strict control over which executables, scripts and macros can be run, and by which user. This guarantees that only those applications that have been previously identified and authenticated will be authorized to run – anything, and everything else known or unknown, will fail to execute.

A Sanctuary Client Driver is installed on each machine that needs protection. This operates at the operating system kernel level. Every time a new file is loaded for execution, the kernel driver intercepts the attempt to load the file into memory, and determines the requesting user's identity, the groups the user belongs to, and the logon session in whose context the call is made.

The kernel driver then proceeds to positively authenticate the executable, script or macro file by means of a cryptographic digest called hash (SHA-1). It is important to emphasize that the authentication takes place when the file is loaded into memory for execution, rather than when the file is read or written to disk.

Once the hash has been calculated, the driver checks whether the current user has been granted the right to run it. If so, then the execution is authorized, if not, access is denied.

Before you activate Sanctuary Application Control Suite

Before protecting your organization against running undesirable executables, you must first:

> Gather a list of executable files that are allowed to run. The system uses a special algorithm to calculate a unique digital signature for each file. You can also import predefined hash lists (Standard File Definitions) of those Windows operating systems supported by Sanctuary to quickly populate the database with all OS files needed.

> Organize these file definitions into logical groups (File Groups), and specify which users/User Groups are authorized to run these files. The relationship here is Application➔File Group➔User/User Group. In large organizations, it is recommended to assign File Groups to User Groups instead of individual users. This has the clear advantage of transferring the administration back to you Windows' user console instead of always using Sanctuary Management Console.

This information is stored in the SecureWave Sanctuary Database.

When a computer signs on to the network

When a computer signs on to the network, the SecureWave Application Server does the following (see *Figure 17*):

> Reads the Security ID (SID) of the machine or account.

> Gets the latest authorizations from the central SecureWave Sanctuary Database (only if its cache is empty or if permissions changed).

> Selects only those parts that changed, compress the list, and signs (or encrypts, depending if you use TLS or not) this information for secure transmission across your LAN or WAN and to avoid tampering.

> Automatically downloads this authorization information to the requesting user/machine.

This authorization information is then stored locally in a secure location on the client's hard disk, where it cannot be tampered with.

When a user asks to run an application

When a user asks to run an application the Windows operating system checks the file extension to determine if it is registered as an executable. Once Windows has determined that it is an executable file (for example, those files with.exe or .dll extension) or is a recognized script or macro file, Sanctuary takes action.

The system checks the entire file at a binary level to calculate a 20-octets hash code, checks it against the list of pre-approved hashes from authorized applications, scripts and macros, and determines whether the file can be run. This verification is transparent to the user and takes place virtually instantaneously.

If the application is on the approved list

If the application is on the approved list the application starts up with no user intervention required. Sanctuary, optionally, logs the successful application access. This feature in not activated by default.

If application access is denied

If application access is denied Sanctuary sends a denial notification to the user and logs the incident. If the local machine has been configured to allow optional override, the user may choose to assume the risk of activating a denied application. This action will be logged as well.

What happens if a computer is taken off the network

Sanctuary is designed to protect computers at all times from running unauthorized programs. The same control and protection is provided to your users even when they are disconnected from the network, for example when laptops are taken off the network. Once a list of hashes has been downloaded, the local copy is used until the computer is reconnected to the network and able to receive automatic updates once again. The local copy is kept in an inaccessible folder and available even when disconnected from the network.

The following schema summarizes all these steps:

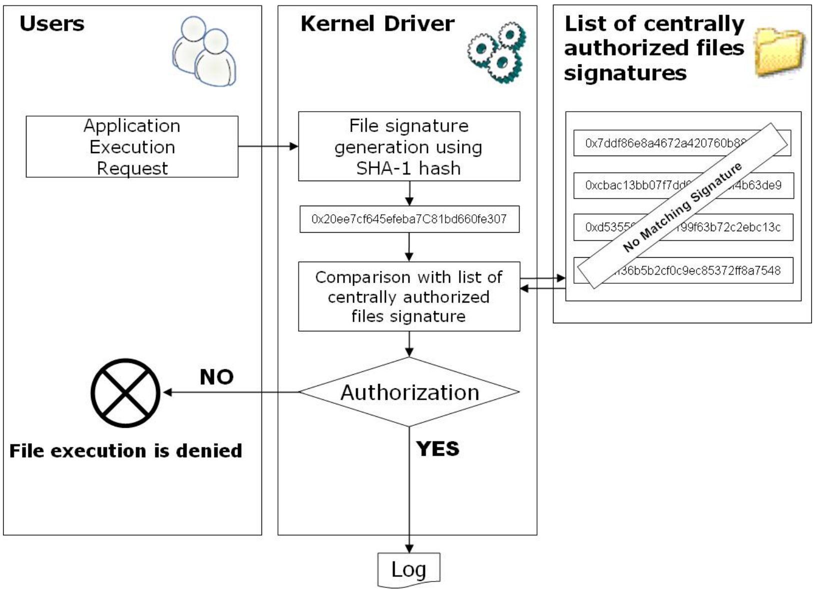

Users
Application Execution Request
Kernel Driver
File signature generation using SHA-1 hash
0x20ee7cf645efeba7C81bd660fe307
Comparison with list of centrally authorized files signature
List of centrally authorized files signatures
0x7ddf86e8a4672a420760b88
0xcbac13bb07f7dd
4b63de9
0xd53556
199f63b72c2ebc13c
136b5b2cf0c9ec85372ff8a7548
No Matching Signature
Authorization
NO
YES
File execution is denied
Log

Figure 15: Sanctuary Application Control Suite authorization process

Sanctuary Device Control

When you first install Sanctuary Device Control default permission rules are created and configured. In addition, devices are **automatically assigned** to predefined device classes according to their Windows classification. The predefined permissions include Copy Limit restrictions and Read/Write permissions for some of the devices.

Even though some users may already be satisfied with these settings, the majority of people prefer to change them to reflect the device policy their organization. Therefore one of the first tasks an administrator does is to change and define new permissions for users, groups, computers, or devices in their network.

Administrators can also manage specific devices by type or brand, if required. They can **assign** rights and attributes by device class, specific device, or specific media to user(s) / user group(s) or to a specific computer.

Before you activate Sanctuary Device Control

Before you activate Sanctuary Device Control, you need to:

> Define your device access policies and decide who can use what and with which restrictions.

> Create rules and permissions using Sanctuary Management Console. Each permission is an association Device Class➔User/User Group. You have several types to choose from: Read, Read/Write, None, Temporary, Scheduled, Copy Limit, Shadow file name or complete content in read/write operations, Offline, Online, etc. In large organizations, we recommend you assign permissions to User Groups instead of individual users. This has the clear advantage of transferring the administration back to you Windows' user console instead of always using Sanctuary Management Console for this job.

The device authorization information is stored in the SecureWave Sanctuary Database.

Communication between the Sanctuary Management Console and the SecureWave Application Server is set to RPC (Remote Procedure Call) level 6. Messages interchanged between them are fully encrypted.

The Sanctuary Management Console connects to the SecureWave Application Server to carry out administrative changes. Therefore, at no time does the Sanctuary Management Console connect directly to the database. All communication with the Database is through and by the SecureWave Application Server(s).

Traffic between the Sanctuary Client Driver and SecureWave Application Server is authenticated based on Private/Public key technology. If you decide to use TLS, the communication is encrypted.

When a computer signs on to the network

You do not have to worry about adding new permissions when an unknown device is connected to a computer in your network. Most devices are declared in one of the Sanctuary Device Control predefined classes during the plug and play discovery phase. Sanctuary Device Control can therefore apply existing device class permissions to the device in most cases. If a device is unknown and does not belong to a predefined device class, the most restrictive permission rule is applied and access is denied until specifically told otherwise. These permissions can even be extended to a specific model installed on a precise computer.

Every time a user wants to access a device, the Sanctuary Device Control driver intercepts the Operating System request at the kernel level. If the device is not in the list of authorized classes and/or specific devices, Sanctuary Device Control will deny its use. If the device is known (e.g., it is in the device class list), the driver checks the user rights in the Access Control List (ACL). In this case, if a user has the right to access a device (for instance a CD burner drive), either Read or Read/Write access is granted. If a user does not have rights on the device, an 'access denied' notification pops up to inform the user — the administrator can optionally define custom messages. The program can log this action, optionally, for the Administrators to analyze.

The following schema summarizes these steps:

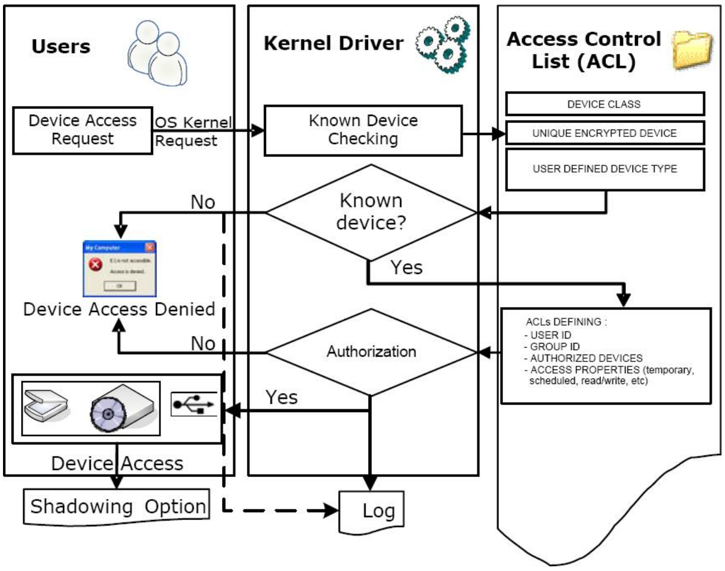


Figure 16: Authorizing a device access

When a user asks to access a device

All computers equipped with the *Sanctuary Client Driver* receive an administrator-created permissions list of all known devices reported by the *Console*. This is forwarded by the *SecureWave Application Server* to the machine. It is delivered in one of several possible ways, depending on whether the computer is or not connected to the network:

Network connection	***Permission updates are done:***
Not available	By importing them from a file.
	Using the list kept internally in the computer's memory.
Available	When the user logs on.
	When the user asks for them using the *Refresh settings* right-click option in the client's system tray.
	When the administrator makes changes and explicitly sends them to a specific computer or all on-line machines.
	If another user logs on.
	Every 60 minutes.
	When communication starts between the Application Server and the client.
	Before triggering a shadow (carbon copy) file transfer.

Table 3: Permissions list updates depending on network connection status

The *SecureWave Application Server*, in turn, communicates with the *SecureWave Sanctuary Database* to retrieve the whole list — only when its cache is empty. The SecureWave Application Server then cryptographically signs the list or encrypts it if a using TLS (Transport Layer Security channel), compresses it, select only those permissions that have changed, and forwards it to the *client computer*.

The process is summarized in *Figure 17*.

If the device is on the approved list

If the device is on the approved list , device access starts with no user intervention required. Sanctuary, optionally, logs the successful access. This feature in not activated by default.

If device access is denied

If device access is denied Sanctuary displays an optional event message to the user and, optionally, logs the incident.

If a computer is taken off the network

Sanctuary Device Control protects all computers, at all times, using the Sanctuary Client Driver. Whenever a computer is disconnected from the network, it is still protected by the permissions that were downloaded from the Sanctuary system when it was last connected. This could be the case with laptop computers. The computer simply accesses its local copy until it is reconnected to the network and able to receive automatic updates once again.

You can create ‘online’ and ‘offline’ permissions for any computer or device on your network, to be applied automatically, as appropriate.

There is no problem if users try to delete or tamper with the list — they simply would not have access at all.

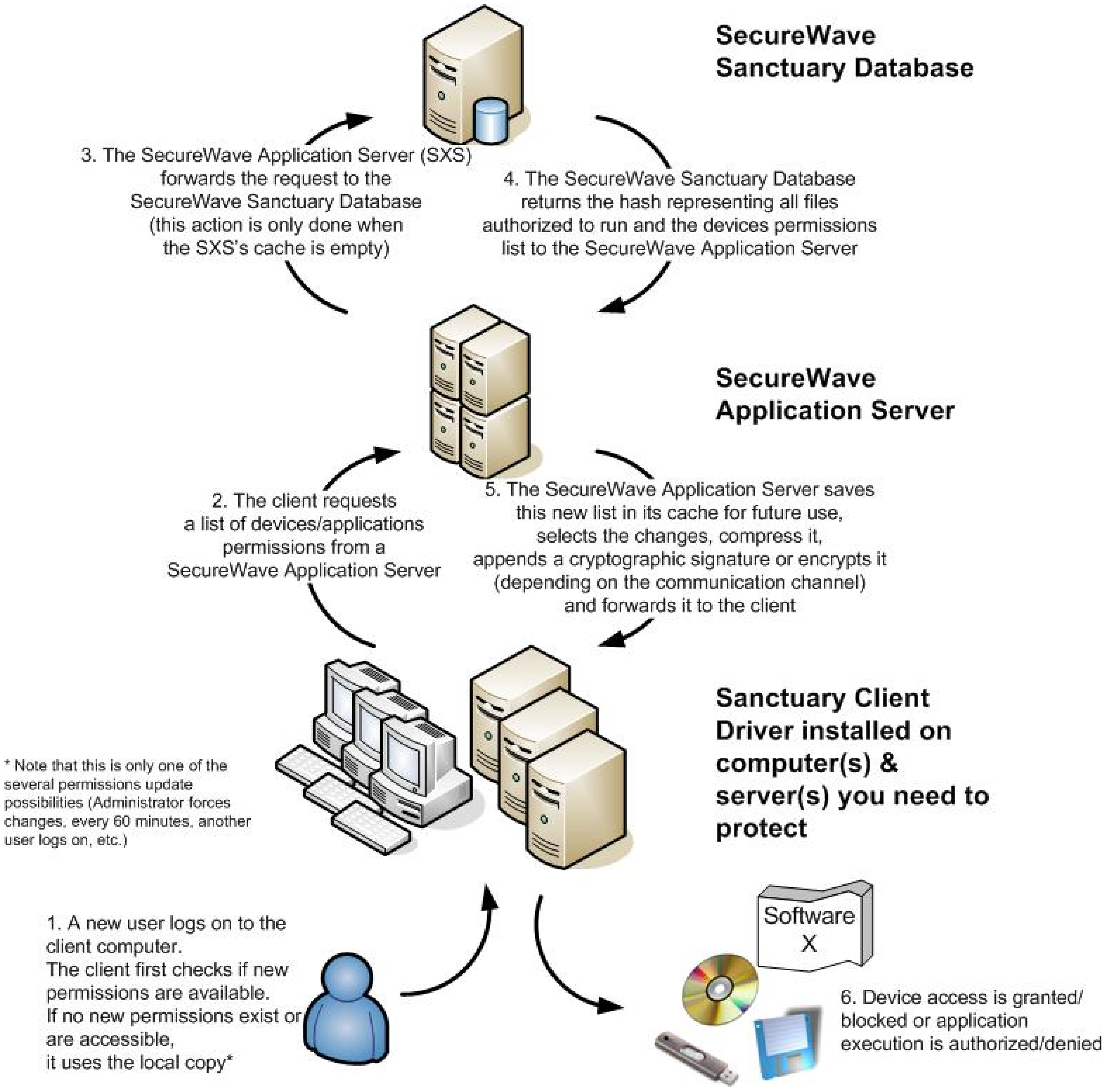

SecureWave
Sanctuary Database
3. The SecureWave Application Server (SXS) forwards the request to the SecureWave Sanctuary Database (this action is only done when the SXS's cache is empty)
4. The SecureWave Sanctuary Database returns the hash representing all files authorized to run and the devices permissions list to the SecureWave Application Server
SecureWave
Application Server
2. The client requests a list of devices/applications permissions from a SecureWave Application Server
5. The SecureWave Application Server saves this new list in its cache for future use, selects the changes, compress it, appends a cryptographic signature or encrypts it (depending on the communication channel) and forwards it to the client
* Note that this is only one of the several permissions update possibilities (Administrator forces changes, every 60 minutes, another user logs on, etc.)
Sanctuary Client Driver installed on computer(s) & server(s) you need to protect
1. A new user logs on to the client computer. The client first checks if new permissions are available. If no new permissions exist or are accessible, it uses the local copy*
Software X
6. Device access is granted/ blocked or application execution is authorized/denied

Figure 17: How the Sanctuary solution works

Glossary

ACL

***A**ccess **C**ontrol **L**ist*. A list that keeps the permissions that each user or group has to a specific system object. Each object has a unique security attribute that identifies which users have access to it.

ADSI

***A**ctive **D**irectory **S**ervice **I**nterface*. Previously known as OLE Directory Services, ADSI makes it easy to create directory management applications using high-level tools such as Basic, Java, or C/C++ without having to worry about the underlying differences between the dissimilar namespaces.

AES

***A**dvanced **E**ncryption **S**tandard*. A symmetric key encryption technique that is replacing the commonly used DES standard. It is the result of a worldwide call for submissions of encryption algorithms issued by NIST in 1997 and completed in 2000.

CAB

File extension for ***cab**inet* files, which are multiple files compressed into one and extractable with the extract.exe utility. Such files are frequently found on Microsoft software distribution disks.

Client Computer

The computers on your network that Sanctuary controls.

CSV

The CSV, ***C**omma **S**eparated **V**alue*, file format allows easy data table retrieval into a variety of applications. It is often used to exchange data between disparate applications. The file format has become a pseudo standard throughout the industry, even among non-Microsoft platforms. Common examples of applications that use this format are spreadsheets and databases. You can also see and edit these files using an ASCII text editor (Notepad, Word, WordPad, Excel, etc.).

Delegation

The act of assign responsibilities for management and administration of a portion of the resources or items used in a shared computing environment to another user, group, or organization.

Dependencies

Additional executable files (.exe, .dll, or others) required by executable files to run properly.

Dependencies are split into two categories: *static dependencies* which are files declared explicitly in the executable file as being required, and *dynamic dependencies* which are additional files an executable may require at runtime.

DN

***D**istinguish **N**ame*. A name that uniquely identifies an object in the Directory Information Tree.

Executable Program

A computer program that is ready to run. The term usually applies to a compiled program translated into computer code in a format that can be loaded in memory and executed by a computer’s processor.

Exploit

A piece of software that takes advantage of a bug, glitch or vulnerability, leading to privilege escalation (exploit a bug) or denial of service (loss of user's services) on a computer system.

File Group

Organizational groups used to cluster authorized executable files. Files must be assigned to 'File Groups' before users can be granted permission to use them. You can choose to assign files to 'File Groups' from various modules throughout the Sanctuary Management Console, e.g. by double-clicking on a file in the *Database Explorer*, *EXE Explorer*, *Log Explorer* or *Scan Explorer.*

GUID

A ***G****lobal* ***U****nique* ***I****dentifier* number generated when the NDS object is created. It is simply an object's NDS attribute. In order to ensure data consistency, Novell eDirectory implements a globally unique ID (GUID) for all objects within the directory. The total number of unique keys (2128 or 3.4028 x 1038) is so large that the possibility of using the same number twice is nearly zero.

Hash

A complex digital signature calculated by Sanctuary Application Control Suite components to uniquely identify each executable file, script or file containing an embedded VBA macro, that can be run. The hash is calculated using the SHA-1 algorithm that takes into account the entire contents of the file.

iFolder

A Novell client that runs on Windows-based computers. It allows a user to work on his files anywhere — online or offline. iFolder integrates encryption and file synchronization services.

LDAP

L*ightweight* ***D****irectory* ***A****ccess* ***P****rotocol*. An LDAP directory entry consists of a collection of attributes and is referenced unambiguously with a name, called a distinguished name (DN). For example, 'cn=Bill Dove: ou=marketing: o=my_company' — 'cn' for common name, 'ou' for organizational units, 'o' for organization. LDAP directory entries feature a hierarchical structure that reflects political, geographic, and/or organizational boundaries.

MAPI

M*essaging* ***A****pplication* ***P****rogramming* ***I****nterface* enables Windows applications to access a variety of messaging systems.

MDAC

M*icrosoft* ***D****ata* ***A****ccess* ***C****omponents*. Required by Windows computers to connect to SQL Server or MSDE databases.

MSDE

M*icrosoft* ***D****ata* ***E****ngine* (also known as Microsoft SQL Server Desktop Engine), is a SQL Server compatible database server, suitable for small and medium size organizations. MSDE databases can subsequently be migrated to SQL Server 2000/2005. SQL Server 2005 Express Edition now supersedes MSDE.

NDAP

N*ovell* ***D****irectory* ***A****ccess* ***P****rotocol*. The NDAP component gives Windows applications full access to the Novell eDirectory and administration capabilities for NetWare servers, and volumes.

NDS

Novell's eDirectory previously called ***N****ovell* ***D****irectory* ***S****ervices*. eDirectory is a hierarchical, object oriented database that represents all the assets in an organization in a logical tree. Assets can include users, positions, servers, workstations, applications, printers, services, groups, etc.

NICI

***N**ovell **I**nternational **C**ryptographic **I**nfrastructure*. NICI is a base set of cryptographic services available for Novell. NICI provides an API set that offers a consistent interface for application developers to use and deploy cryptography within their applications.

OU

***O**rganizational **U**nits*. A part of the Active Directory (AD) structure inherited from Novell's NDS structure. Within Novell's NDS/eDirectory there are three classes of objects in the NDS batabase: Roots, Containers, and Leafs. There are three supported types of container objects: Country (C=), Organizations (O=), and Organizational Units (OU=).

Private Key

One of two keys used in public key encryption. The sender uses the private key to create a unique electronic number that can be read by anyone possessing the corresponding public key. This verifies that the message is truly from the sender.

Public Key

One of two keys in public key encryption. The user releases this key to the public, who can use it for encrypting messages to be sent to the user and for decrypting the user's digital signature.

RPC

A ***R**emote **P**rocedure **C**all* is a protocol that allows a computer program running on one host to run a subroutine on another host. RPC is used to implement the client-server model of distributed computing.

RSA Encryption

In 1977, Ron Rivest, Adi Shamir, and Len Adleman developed the public key encryption scheme that is now known as RSA, after their initials. The method uses modular exponentiation, which can be performed efficiently by a computer, even when the module and exponent are hundreds of digits long.

SFD

SecureWave provides a number of pre-computed file hashes for most versions of suites and Windows Operating Systems, in several languages, and for all the available Service Packs. The file hashes are referred to as *Standard File Definitions* or SFD. They are installed during the setup, but you can import them as soon as SecureWave releases new ones. You can find the latest ones on our Web site.

SHA-1

***S**ecure **H**ash **A**lgorithm 1*, as defined in the Federal Information Processing Standards Publication 180-1. This algorithm produces a one-way 160-bit hash that can be used for a variety of applications including authentication and cryptography.

SID

***S**ecurity **id**entifier*, a security feature of Windows NT and 2000 operating systems. The SID is a unique name (alphanumeric character string) used to identify an object, such as a user or a group of users in a network.

Windows grants or denies access and privileges to resources based on an ACL (***A**ccess **C**ontrol **L**ist*), which uses a SID to uniquely identify users and their group memberships. When a user requests access to a resource, the user's SID is verified by the ACL to determine if the user, or the group he belongs to, is allowed to perform that action.

SQL

***S**tructured, **Q**uery **L**anguage*, a language used to construct database queries.

SUS

***S**oftware **U**pdate **S**ervices* is a tool provided by Microsoft to assist Windows administrators with the distribution of security fixes and critical update releases.

SecureWave Application Server

The main component of all Sanctuary’s products. Beside calculating hashes, authorizing applications and devices, it serves as a bridge between the database and the client.

TCP/IP

***T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol*. The protocol used by the client computers to communicate with the SecureWave Application Server.

TLS

***T**ransport **L**ayer **S**ecurity*. The Transport Layer Security (TLS) protocol (based on SSL — Secure Socket Layers) addresses security issues related to message interception during communication between hosts. The deployment of TLS, client and server side, is the primary defense against compromised clients or mixed networks where is possible to intercept transmitted messages.

VBScript

A scripting language created by Microsoft embedded in many applications used in Windows. Although it allows for powerful interoperability and functionality, it also creates a great deal of security risks unless it is tightly controlled.

Vulnerability

A weakness or other kind of opening in a system, usually caused by a bug or other design flow.

Well-Known Security Identifiers

A security identifier (SID) is a unique value used to identify a security principal or security group. The values of certain SIDs remain constant across all installations of Windows systems and for this reason are termed well-known SIDs. Everybody, Local, Guest, Domain Guest, etc. are some examples of SIDs.

WMI

***W**indows **M**anagement **I**nstrumentation*. WMI is a standard technology to access management information in an enterprise environment. WMI uses the Common Information Model (CIM) industry standard to represent systems, applications, networks, devices, and other managed components. You can use WMI to automate administrative tasks in an enterprise environment. WMI improves administrative control by allowing administrators to correlate data and events from multiple sources and vendors on a local or enterprise basis. It is used as a complement to ADSI.

WSUS

***W**indows **S**erver **U**pdate **S**ervices* (previously SUS v2.0) is a new version of Software Update Services (SUS).

Zero-Day exploit

A zero-day exploit is a malicious code that takes advantages of a security vulnerability on the same day this vulnerability is known. Since the vulnerability is not known in advance, there is no way to guard against the exploit before it happens if you are using traditional solutions (e.g. blacklist antivirus programs).

Index of Figures

Index of Tables

Index