LynxSecure 6.3.0 Development System Introduction

LynxSecure Release 6.3.0-rev16326

Contents

List of Figures

List of Tables

Preface

About this Guide

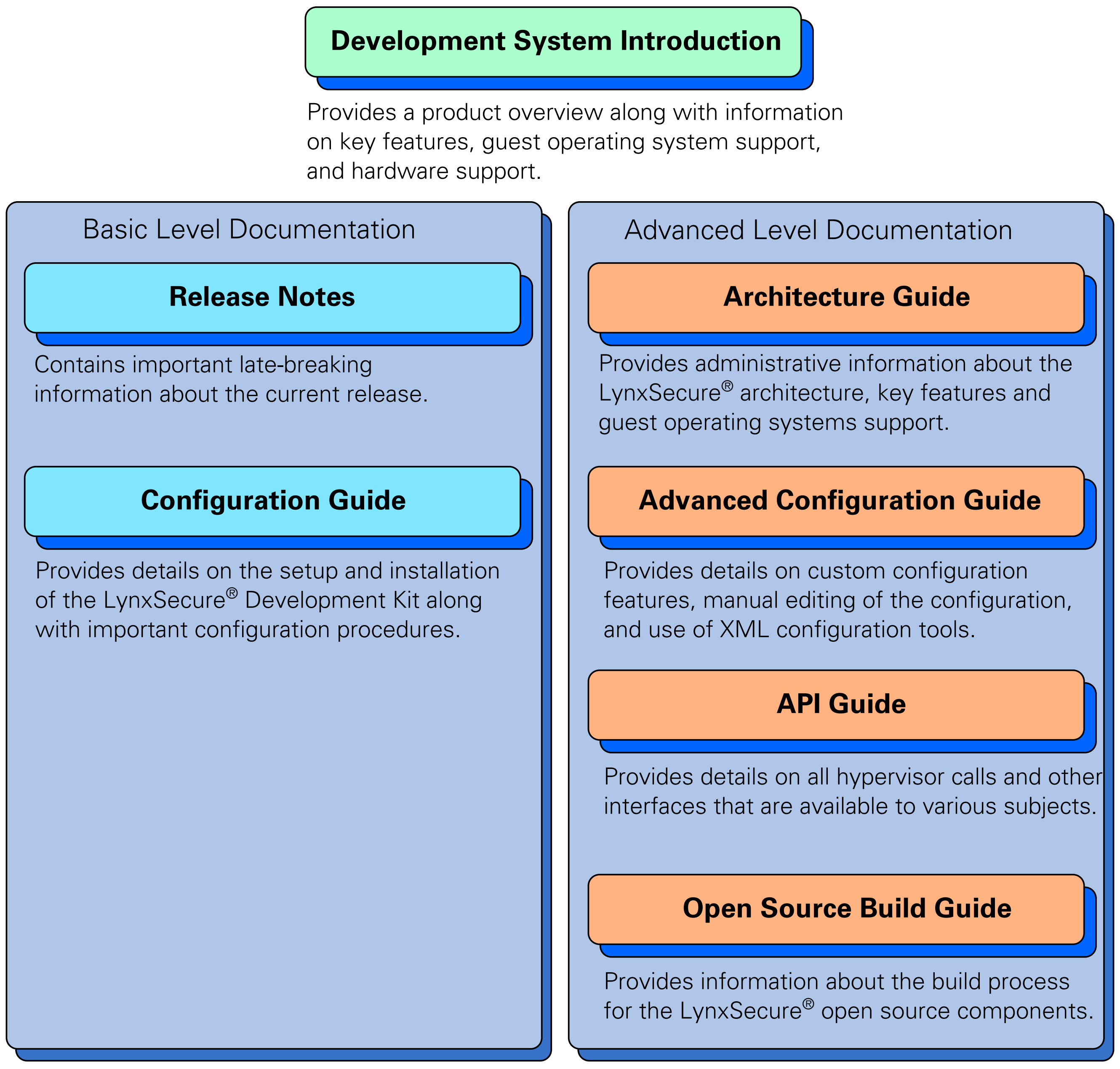
This document, *LynxSecure® Development System Introduction*, offers an overview of the LynxSecure operation and the development workflow.

Intended Audience

The information in this guide is designed and written for system integrators and software developers who will build applications on top of LynxSecure and the available subjects. A basic understanding of Linux®/Unix® and familiarity with fairly complex configuration procedures are recommended.

For More Information

For more information on the features of LynxSecure, refer to the following printed and online documentation.


Development System Introduction
Provides a product overview along with information on key features, guest operating system support, and hardware support.
Basic Level Documentation
Release Notes
Contains important late-breaking information about the current release.
Configuration Guide
Provides details on the setup and installation of the LynxSecure® Development Kit along with important configuration procedures.
Advanced Level Documentation
Architecture Guide
Provides administrative information about the LynxSecure® architecture, key features and guest operating systems support.
Advanced Configuration Guide
Provides details on custom configuration features, manual editing of the configuration, and use of XML configuration tools.
API Guide
Provides details on all hypervisor calls and other interfaces that are available to various subjects.
Open Source Build Guide
Provides information about the build process for the LynxSecure® open source components.

Typographical Conventions

The typefaces used in this manual, summarized below, emphasize important concepts. All references to file names and commands are case sensitive and should be typed accurately.

Font and Description	Examples
Times New Roman 10 pt. - Used for body text; *italicized* for emphasis, new terms, and book titles.	Refer to the *LynxSecure User's Guide*

Courier New 9 pt. - Used for environment variables, file names, functions, methods, options, parameter names, path names, commands, and computer data. Commands that need to be	`ls -l`
	`myprog.c`
	`/dev/null`
	`login:` **`myname`**

Font and Description	Examples
highlighted within body text, or commands that must be typed as-is by the user are **bolded**.	`# `**`cd /usr/home`**

Courier New Italic 9 pt. - Used for text that represents a variable, such as a file name or a value that must be entered by the user.	`cat` *`filename`*
	`mv` *`file1 file2`*

Courier New 7 pt. - Used for blocks of text that appear on the display screen after entering instructions or commands.	`Kernel: target1.srp`
	`> Loading: target1.srp`
	`> ......................`
	`> Booting`

Univers 45 Light Bold 8 pt. - keyboard options, button names, and menu sequences.	Enter, Ctrl-C

Special Notes

The following notations highlight any key points and cautionary notes that may appear in this manual.

NOTE: These callouts note important or useful points in the text.

CAUTION! Used for situations that present minor hazards that may interfere with or threaten equipment/performance.

Technical Support

Lynx Software Technologies, Inc. Technical Support is available Monday through Friday (excluding holidays) between 8:00 AM and 5:00 PM Pacific Time (U.S. Headquarters) or between 9:00 AM and 6:00 PM Central European Time (Europe).

The Lynx Software Technologies, Inc. World Wide Web home page [http://www.lynx.com] provides additional information about our products.

Lynx Software Technologies, Inc. U.S. Headquarters

Internet: `<support@lynx.com>`
Phone: (408) 979-3940
Fax: (408) 979-3920

Lynx Software Technologies, Inc. Europe

Internet: `<tech_europe@lynx.com>`
Phone: (+33) 1 30 85 06 00
Fax: (+33) 1 30 85 06 06

CHAPTER 1 Introduction

About LynxSecure

LynxSecure®is a real-time development platform that leverages multi-core CPU hardware virtualization features to enhance OEM embedded solutions with accelerated performance and security property enforcement. It is primarily targeted to raise the assurance of systems that perform critical computing functions in regulated environments. Common use cases include: separating critical apps from internet domains, isolating security functions from application domains, verifying and filtering inter-domain communication. LynxSecure executes underneath applications and operating systems, runs completely transparent and is non-bypassable and tamper-proof. The software can be embedded into a broad class of devices from embedded devices, IoT gateways as well as IT platforms like desktop, laptops and servers.

Least Privilege Architecture for Building Secure Systems

LynxSecure upholds the principles of least privilege featuring limited kernel space functionality, lightweight simple design, and explicit granular authorization of all system control functions. Unlike traditional OS and hypervisor kernels that include drivers, I/O stacks, and application APIs, the LynxSecure separation kernel exports all I/O and application support in user space. Instead LynxSecure limits its kernel space functionality to resource partitioning, controlling data flow between partitions, and mediating access to system state change functions. This provides a robust foundation for the development of high assurance systems. The LynxSecure separation kernel provides the foundational safety and security properties to host scalable, high performance, and completely secure architectures.

Platform Integrity with LynxSecure

The LynxSecure SDK offers advanced resource, scheduling, and security controls that exceed traditional operating systems and micro-kernel offerings. These granular controls allow developers to explicitly define how a computing platform executes with traceable evidence - from specification to instantiation, establishing platform integrity for the following design patterns:

- Safety and Security Domain Isolation
- Trusted Execution Environments
- Reference Monitor Plugins
- Firewalls, IDS, Encryption, Guards

Virtualization of Guest Operating Systems

The use of LynxSecure virtualization technology allows multiple types of operating systems to share a single physical hardware platform. Virtualization technology allows for significant cost savings through hardware consolidation, while retaining the ability to leverage the ecosystem of applications that belong to different operating system domains into a single platform.

Highly Scalable Technology

LynxSecure provides a scalable solution ranging from deeply embedded systems to high-end workstations and servers for the design of applications in embedded avionics products, weapons systems, and critical infrastructure control systems.

High Performance and Real-Time

LynxSecure is designed to securely support the latest capabilities of Intel® 64 bit CPUs. Extensive research and analysis went into the design of multi-core, CPU virtualization, memory management, I/O device control, and interrupt management to ensure that hardware is utilized to its fullest extent to enhance protection while still provide high performance computing capabilities. The following features contribute to LynxSecure's high performance capabilities:

64-bit and 32-bit virtualization support

Hosts 32-bit and 64-bit guest OSs and allocates up to 96GB of RAM for each guest OS.

Multi-core virtual CPU support

Supports symmetric multi-processing (SMP) guest OSs.

Multi-core physical CPU support

Allocates all available target platform CPU to host guest OSs and LynxSecure bare-metal applications.

Intel Hyper Threading Support

Supports Intel CPU core Hyper Threads to maximize processing throughput on each physical CPU core.

Intel VT-x direct execution

Uses Intel VT-x to allow guest OSs to execute directly on CPU cores to reduce virtualization overhead.

Intel VT-d direct device assignment

Uses Intel VT-d to allow guest OSs and LynxSecure applications to directly control and isolate communication with physical devices.

Intel EPT

Uses Intel Extended Page Tables to allow guest OSs to manage virtual memory page tables without the assistance of the hypervisor resulting in near native guest OS memory performance.

Intel PAT

Allows users to simply configure the caching attributes of memory regions used by subjects running on LynxSecure to increase performance or mitigate timing channel attacks of certain applications.

Real-time Scheduler

Features a real-time programmable cyclic schedule that ensures all system operations are executed on clock tick precision to mitigate denial of service attacks and covert timing channels.

High Assurance

The technology was designed to satisfy high assurance computing requirements in support of the NIST, NSA Common Criteria, and NERC CIP evaluation processes which are used to regulate military and industrial computing environments in key markets such as Industrial, Automotive, Medical, Defense, Aerospace and Cyber Security.

The LynxSecure Development System

The LynxSecure Development System is a complete offering of software, technical training, and optional Commercial Off-The-Shelf (COTS) hardware used to provide system engineers the tools necessary for building advanced cost savings and security enhanced solutions based on LynxSecure virtualization technology.

The development system software includes both the LynxSecure Separation Kernel along with the development tools which allow engineers and architects the ability to configure and deploy LynxSecure onto their target platforms. Additionally, the development tools include software libraries that allow developers to create custom software that interacts with LynxSecure to add additional levels of security and reliability functions to their system designs.

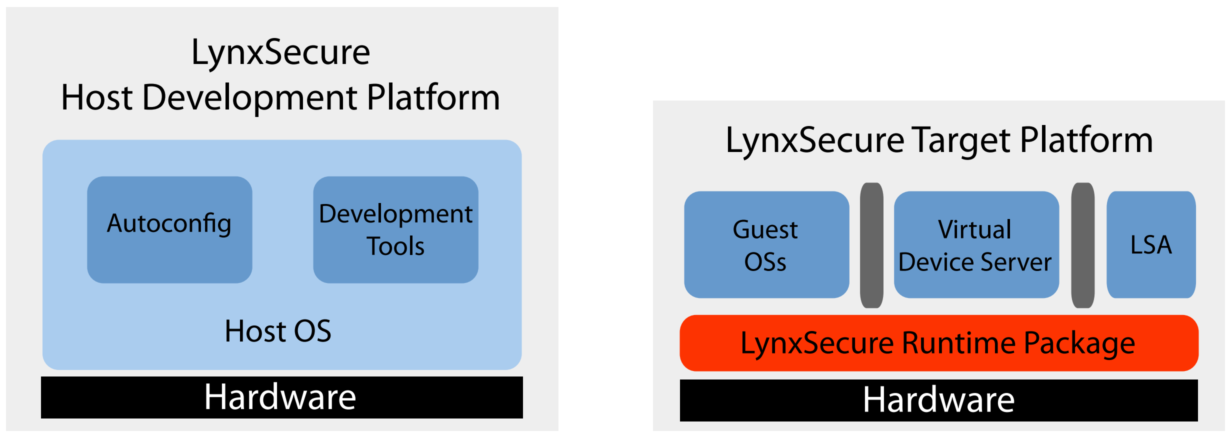
COTS hardware it offered as an option with the LynxSecure development tools which is pre-configured on a reference host development system along with an example demonstration of a LynxSecure Separation Kernel deployment for customer preferred target platforms to get customers started quickly.

Major Components

The LynxSecure development system includes a large collection of software components for developing and deploying virtualization-based solutions. These software components that can be categorized into the following groups:

1. Host Development Software — software used for developing, configuring, and deploying LynxSecure on target platforms.
2. Target Platform Software — software deployed on the target systems.

Figure 1-1: LynxSecure Development System Software Components

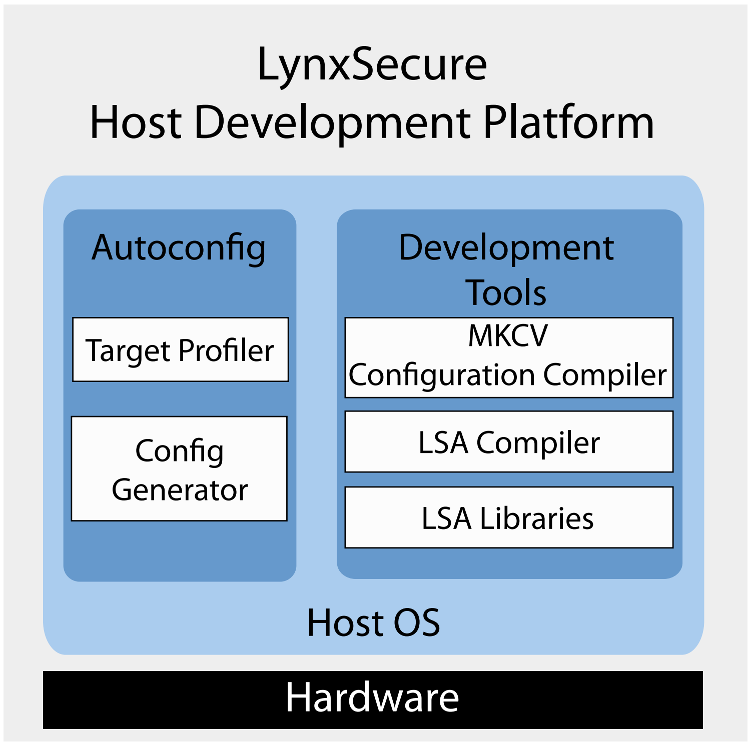


CHAPTER 2 *LynxSecure Host Development and Target Platform*

This chapter provides information on the components and subcomponents contained within the LynxSecure® Host Development Platform and the LynxSecure Target Platform.

LynxSecure Host Development Platform

Figure 2-1: LynxSecure Host Development Platform



LynxSecure Host Development Platform Components

The following information outlines the Development Components necessary for system engineers and architects to configure and compile custom host virtualization solutions.

Autoconfig

Autoconfig is a Command Line Interface (CLI) tool used to define a target system's configuration. It is used to configure multiple numbers of Guest Operating Systems to manage the CPU and memory resource allocation per Guest Operating System and assign I/O peripherals to each. Autoconfig outputs the configuration to an XML file format and forwards that configuration to the policy compiler in order to generate a LynxSecure runtime image.

Autoconfig — Target Profiler

The Autoconfig Target Profiler is a sub-component of the Autoconfig CLI tool. The Autoconfig Target Profiler collects hardware information from target hardware and creates the hardware profile entry in a local database to assist in the configuration process.

Autoconfig — Configuration Generator

The Autoconfig Configuration Generator is a sub-component to Autoconfig. It parses shorthand configuration specification syntax to generate detailed XML policy specifications. The Autoconfig Configuration Generator automates the complexities of calculating low-level resource assignment abstractions for the user. Users can override the Autoconfig Configuration Generator by manually editing the XML specification to explicitly define values such as physical memory addresses.

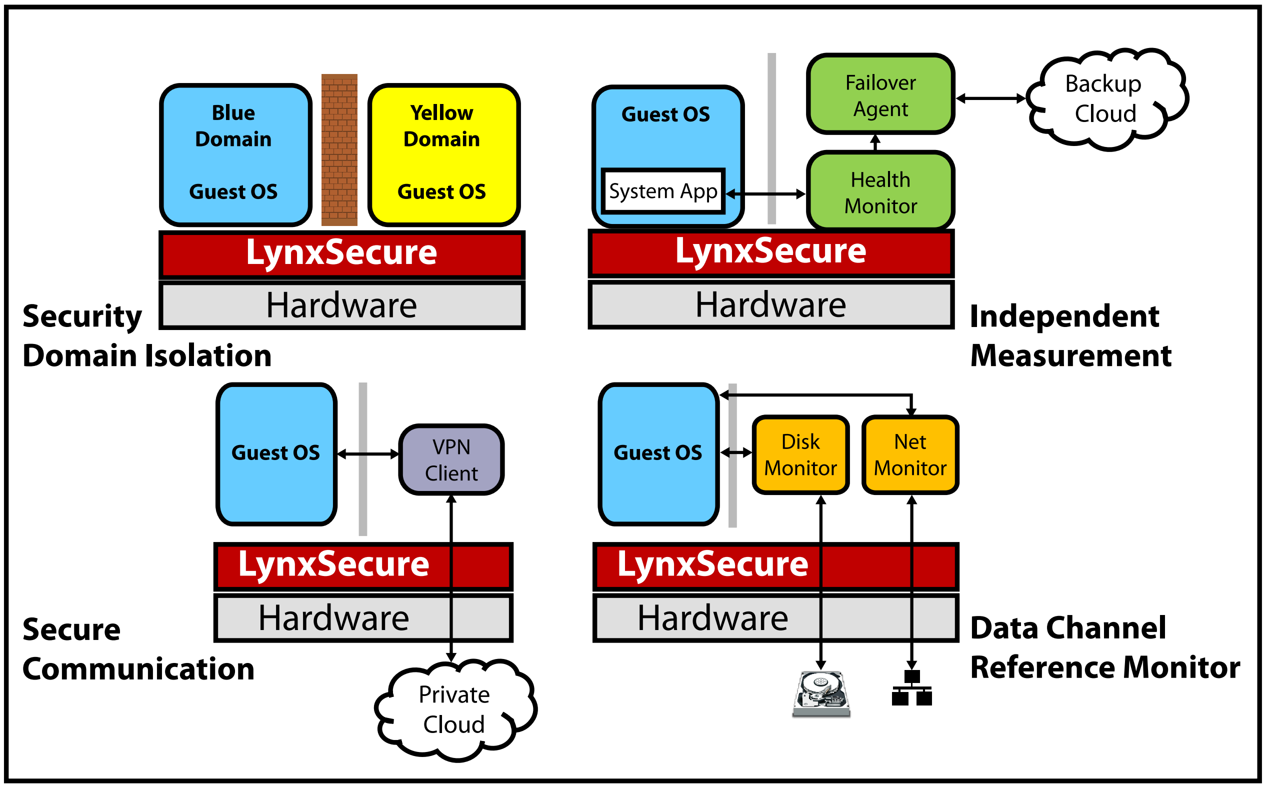
MKCV — Configuration Policy Compiler

The MKCV Configuration Policy Compiler assembles the XML configuration specification and creates an immutable binary configuration policy that controls the functional behavior of the Separation Kernel during runtime.

LynxSecure Trusted Application Library & Compiler

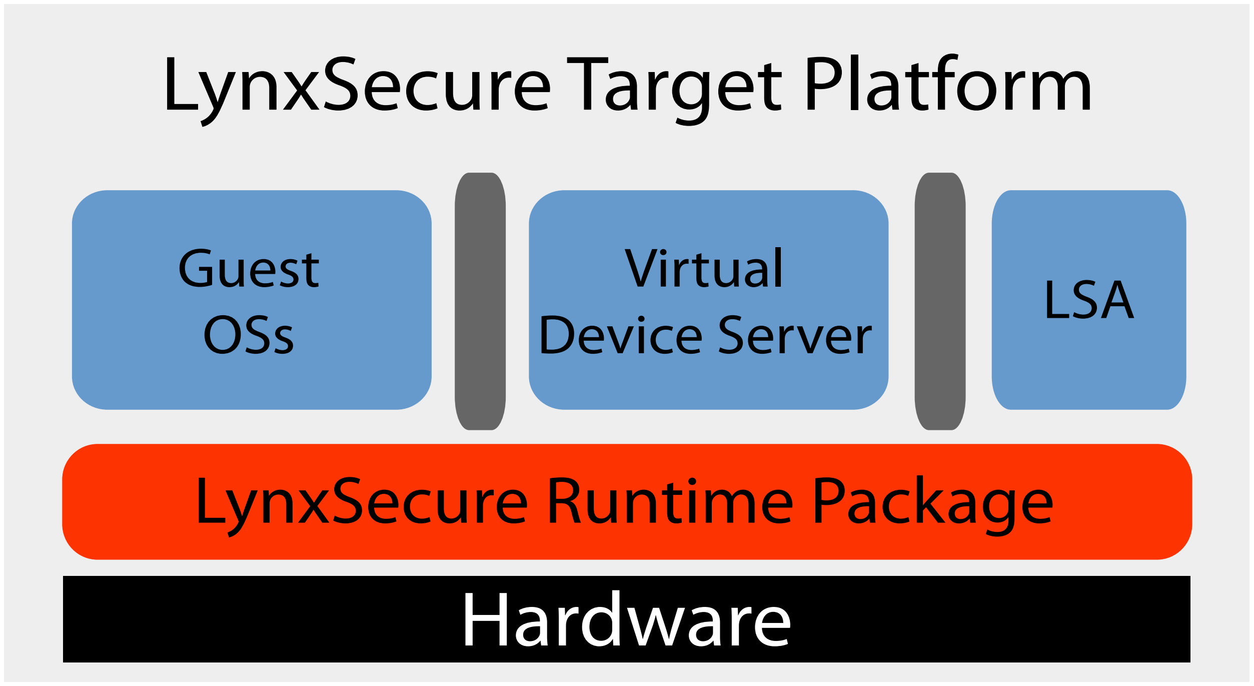
The LynxSecure Trusted Application Library is a set of C libraries and compilers that enable developers to create custom software that can run in tandem with Guest Operating Systems and interact with the Separation Kernel to enhance security and reliability of the Target System.

Figure 2-2: LynxSecure Trusted Application Custom Solution Examples



LynxSecure Target Platform

Figure 2-3: LynxSecure Target Platform



LynxSecure Target Platform Components

The following information outlines the Target Components necessary for system engineers and architects to create target virtualization solutions.

LynxSecure Runtime Package

The LynxSecure Runtime Package is the binary image generated by the MKCV compiler. It is the component that virtualizes Guest Operating Systems, LynxSecure Applications, and enforces resource isolation and information flow control throughout the system. This user-defined process allows for the ultimate in flexibility.

Virtual Device Server

The Virtual Device Server a Guest Operating System that is specifically constructed for the development system to allow other Guest Operating Systems to share physical devices and perform management services.

LynxSecure Application (LSA)

The LynxSecure Application (LSA) is a single threaded process which runs directly on a CPU core without the assistance of a Guest Operating System. LSAs are designed to perform critical functions that require minimal attack surface and deterministic behavior.

CHAPTER 3 *LynxSecure Separation Kernel Technology*

This chapter provides information on the LynxSecure® Separation Kernel Technology; a new software virtualization technology also known as a "Separation Kernel Hypervisor". The following information describes the unique design approach LynxSecure takes in creating secure virtual environments.

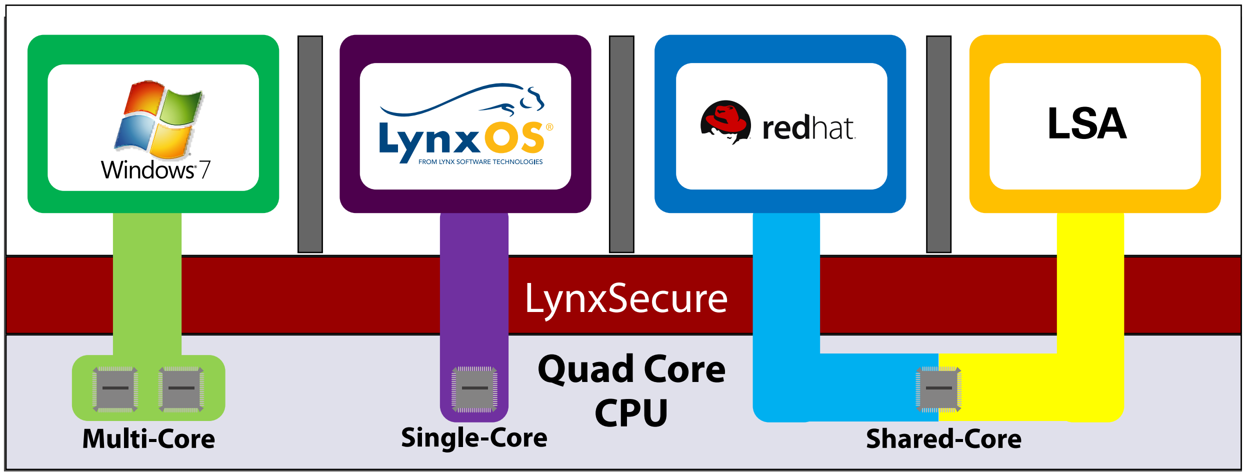
LynxSecure Separation Kernel Hypervisor

The LynxSecure Separation Kernel Hypervisor is a virtualization technology designed to support embedded, real-time, safety critical, and security critical solutions.

The core foundation of LynxSecure is the Separation Kernel. The Separation Kernel is a small executable that runs at the highest CPU privilege to control the hardware resources that is defined with the LynxSecure development tools. The Separation Kernel is very unique in that it does not include common kernel level software components such as file systems, device drivers, I/O stacks, or virtualization services. LynxSecure is designed in a manner where common kernel components run independently in either the Guest Operating Systems or in the De-privileged Virtualization Layer to reduce size, complexity, and overall trusted code base of the privileged kernel.

The LynxSecure Separation Kernel provides the ability to assign one or more CPU cores to Guest Operating Systems and provide the ability to share a CPU core between multiple Guest Operating Systems for optimal CPU allocation.

Figure 3-1: LynxSecure CPU Allocation



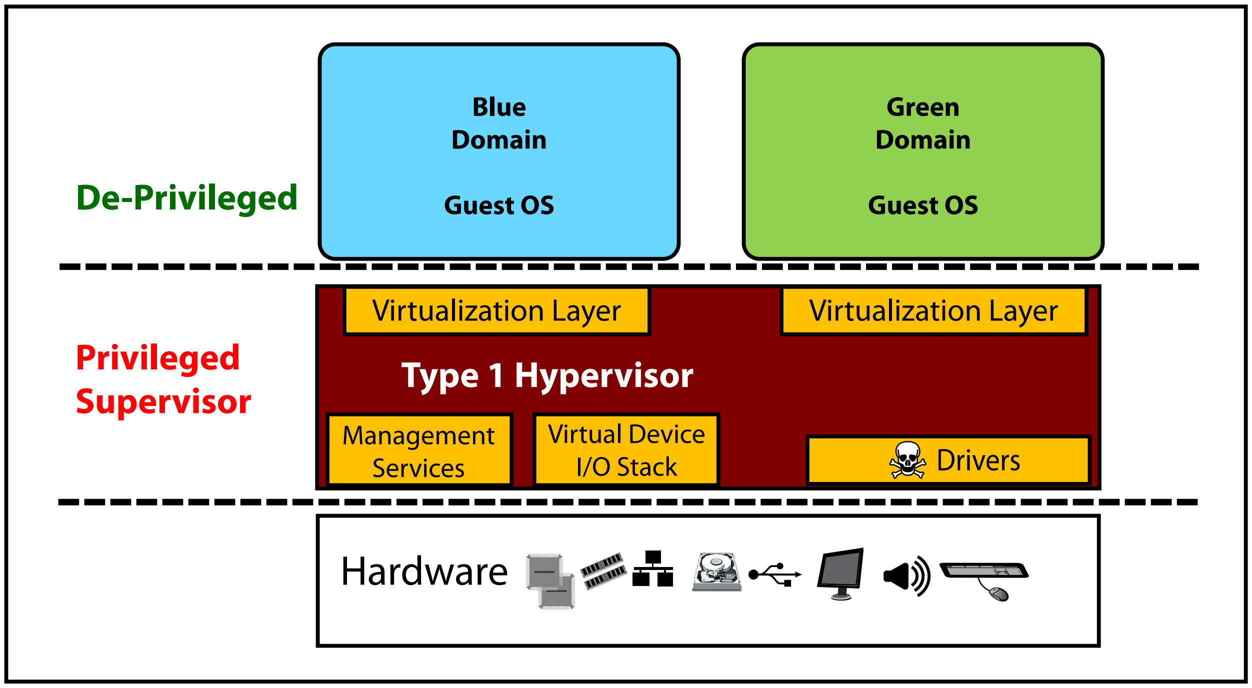
The LynxSecure Separation Kernel is designed for use in defense systems for the purpose of securely consolidating multiple computing platforms that possess different security levels. It is designed with a path of "least resistance" approach that explicitly controls hardware platform resources to create secure computing zones for Guest Operating Systems and "bare-metal" applications. The result is a convenient infrastructure that provides availability, integrity, and confidentiality for each zone that is independently protected from one another.

LynxSecure Type Zero Virtualization Architecture

LynxSecure Type Zero Virtualization Architecture is a new bare-metal architecture designed by Lynx Software Technologies, Inc. It differentiates itself from Type-1 Hypervisors by removing as much functionality from the kernel's trusted supervisor mode of execution and exporting the functionality in partitioned untrusted modes of execution.

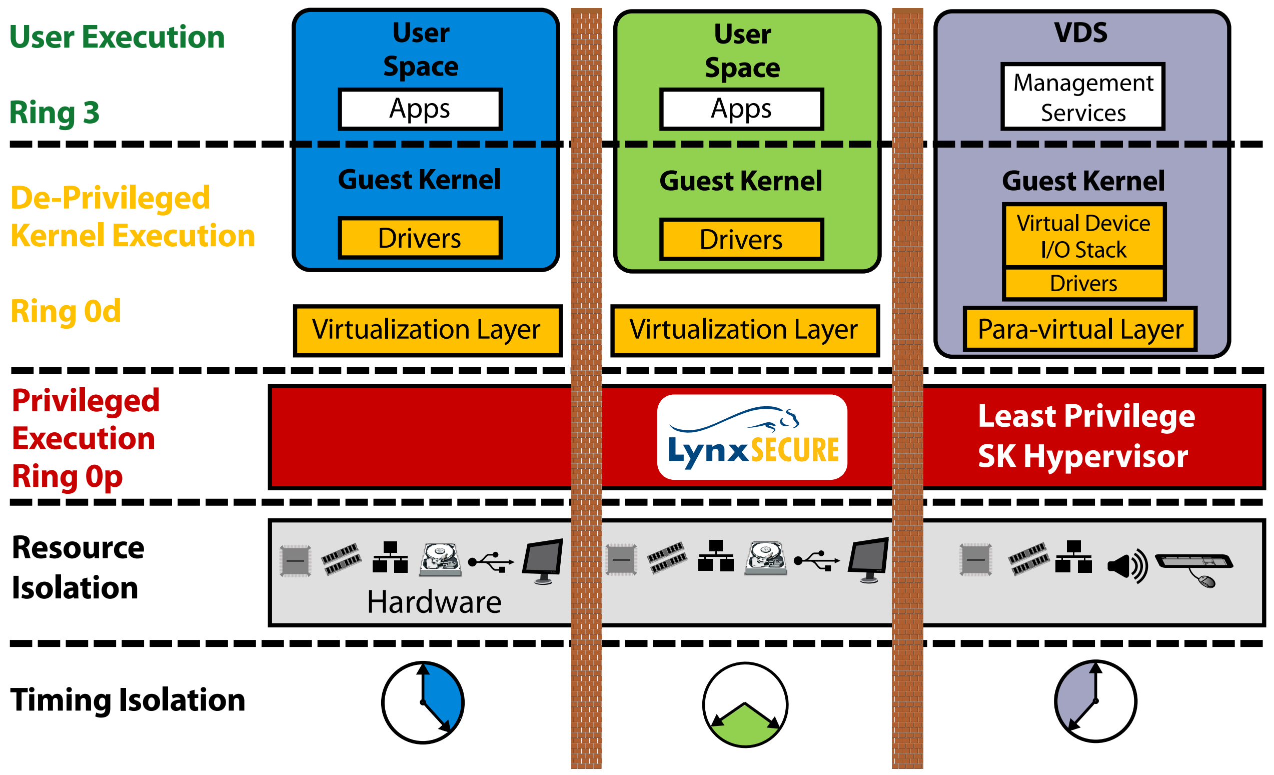
The generic architecture of a Type-1 Hypervisor's virtualization model is an embedded operating system that hosts the software required to virtualize Guest Operating System environments using CPU virtualization, virtual networking, virtual storage, and management services.

Figure 3-2: Type-1 Virtualization Model



LynxSecure Type Zero Virtualization Architecture provides similar functions as Type-1 Hypervisors, but runs the majority of the virtualization logic in separate execution environments to reduce the amount of resources for trusted logic that possesses supervisory control over hardware and user applications.

Figure 3-3: LynxSecure Type Zero Virtualization Model



LynxSecure Virtualization Layer

The LynxSecure Virtualization Layer is an independent software component that runs underneath each fully virtualized guest OS, and runs external to the Separation Kernel. The Virtualization Layer provides ability to emulate full PC platforms and devices underneath Guest Operating System where the Guest Operating Systems are able to share platform resources without installing special drivers. The LynxSecure Virtualization Layer is specifically designed to run outside of the Separation Kernel so that complex platform emulation code does not have privileged access to the hardware resources.

LynxSecure achieves Operating System virtualization through a unique design that differs greatly from traditional Type-1 and Type-2 Hypervisors.

LynxSecure Major Capabilities

The following information describes some of the LynxSecure Major Capabilities.

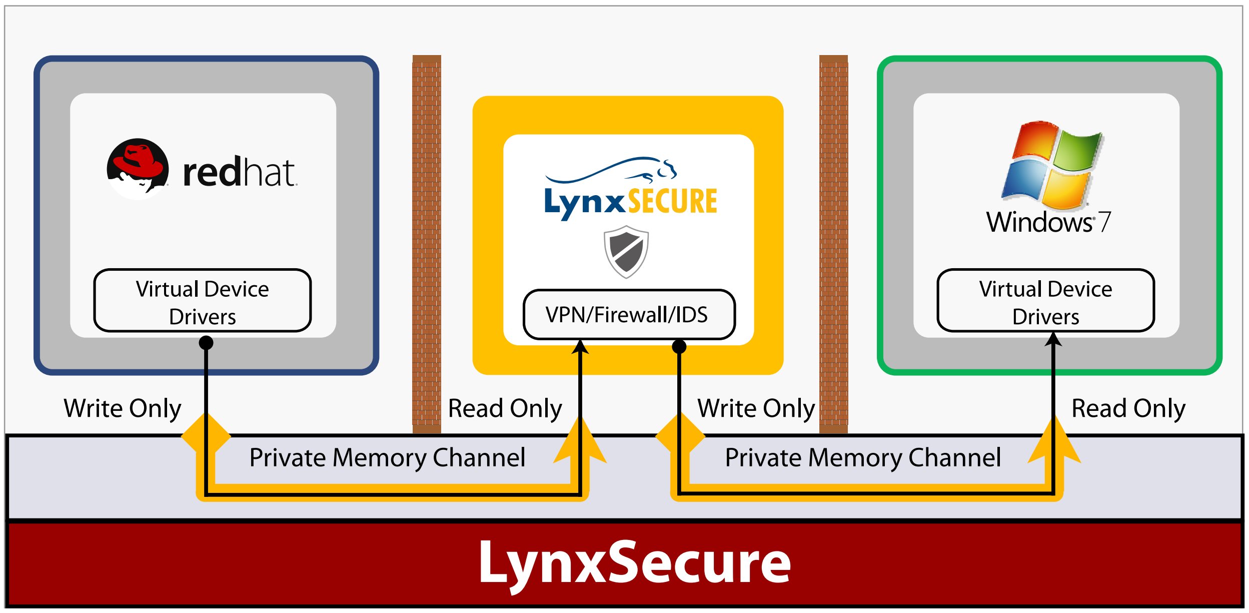
Full Guest Operating System Virtualization

Full Guest Operating System Virtualization provides the ability to simultaneously run multiple independent operating systems on the same hardware platform. Guest Operating System virtualization reduces the cost of dedicating a hardware platform to every Guest Operating System. Guest Operating System virtualization also allows Guest Operating Systems to be transferred to other physical hardware platforms running the Type Zero Hypervisor, which greatly improves that ability to perform hardware maintenance and upgrades.

LynxSecure Applications

LynxSecure Applications are built from tools using the bare-metal approach that utilize CPU cores without relying on the assistance of a Guest Operating System. This bare-metal approach is extremely useful for computing environments (e.g. cryptography key management) that need to be isolated from untrustworthy operating systems. By removing the dependencies from an operating system, verification becomes drastically easier. It also allows highly trusted components to co-exist with general purpose Guest Operating Systems, thereby creating a security posture that is vastly superior to non-virtualized solutions.

Figure 3-4: LynxSecure Application with Data Flow Control



Guest Operating System CPU Scheduling

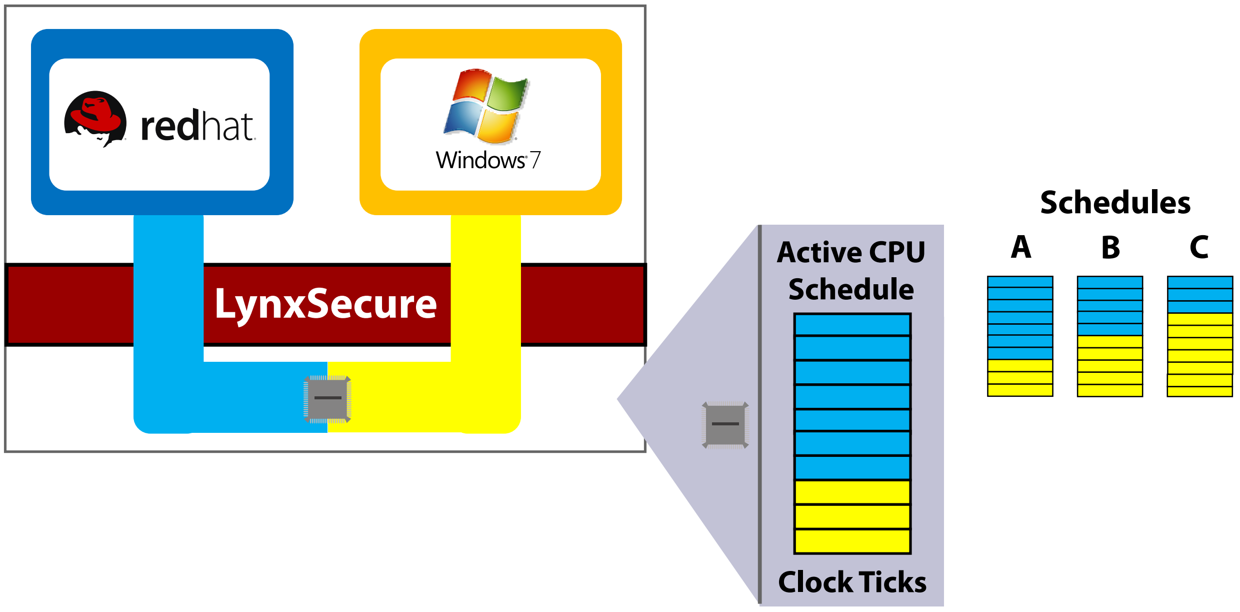
When more than one Guest Operating System shares one or more CPU cores, LynxSecure provisions users with precise control on time duration Guest Operating System execution with a cyclic real-time scheduler. Users can create multiple execution schedules that can be dynamically loaded at runtime to accommodate various use cases regarding load balancing.

The CPU core is prepared with three alternate execution schedules:

Table 3-1: Shared CPU Core Scheduler Values

Schedules	Guests
A	Guest 1 = 75% Guest 2 = 25%
B	Guest 1 = 50% Guest 2 = 50%
C	Guest 1 = 25% Guest 2 = 75%

Figure 3-5: Shared CPU Core Scheduler



LynxSecure Memory Allocation

LynxSecure Memory Allocation provides architects explicit control over system memory to be assigned to Guest Operating Systems, LSAs, and communication channels. LynxSecure guarantees that memory resources can only be accessed by predefined authorized subjects.

Management Guest Operating System Virtual Device Server

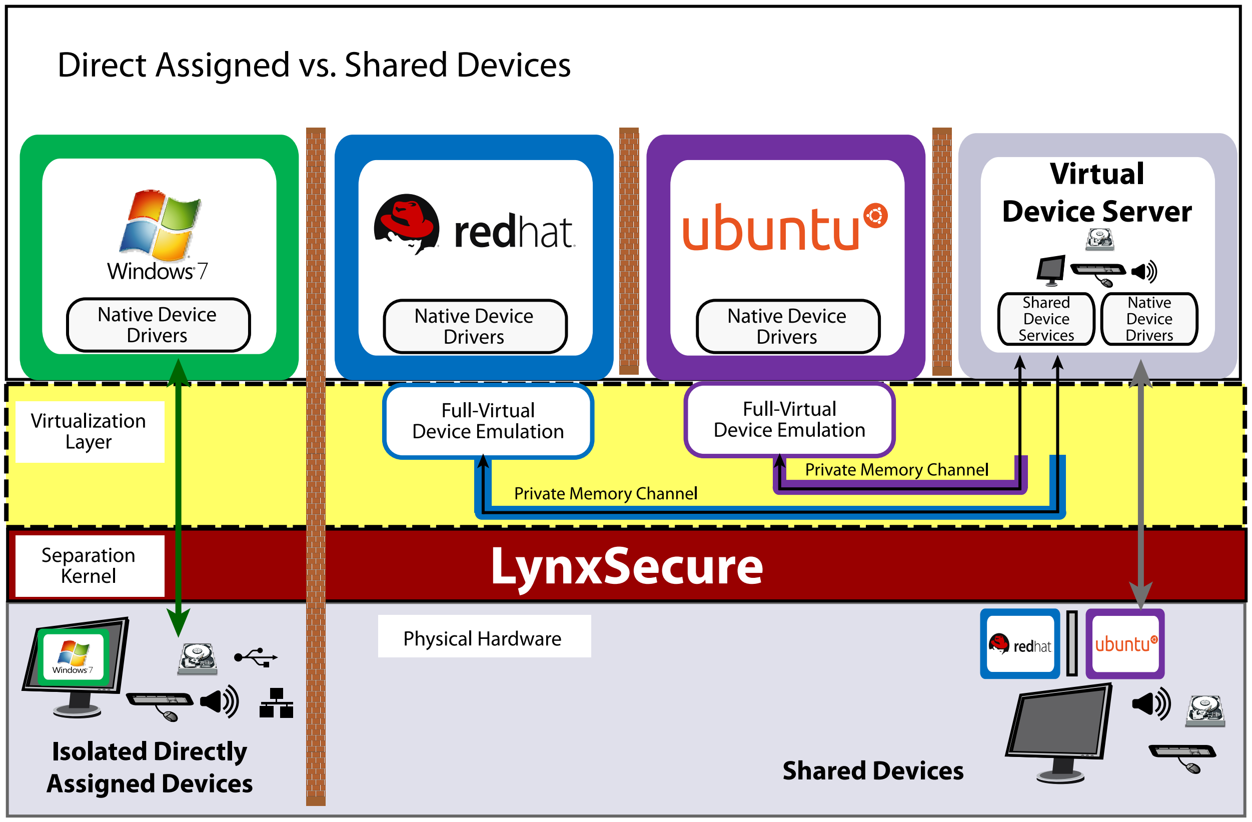
The LynxSecure Development System supplies dedicated Guest Operating System prepackaged with software services for virtualization solutions that require remote administration features and the ability for Guest Operating Systems to share resources such as network interface cards, storage devices, and data files.

Guest Operating System Virtual I/O

LynxSecure provides two methods of Guest Operating System Virtual I/O that allow Guest Operating Systems to I/O peripherals:

- Direct Assignment
- Virtual Device Emulation

Figure 3-6: Virtual I/O – Direct Assignment vs. Shared Device Emulation



Direct Device Assignment

Direct Device Assignment is a customizable feature that gives a Guest Operating System the ability to directly control an external I/O device using its native drivers. Direct device assignment is critical for embedded architectures that need the maximum performance out of I/O devices.

Virtual Device Emulation

Virtual Device Emulation is a device sharing mechanism provided by the LynxSecure Virtualization Layer. The LynxSecure Virtualization Layer provides the option to present an independent virtual device interface to the Guest Operating Systems. The Guest Operating Systems connect to the emulated devices using standard device drivers included in the Guest Operating System. The emulated devices then connect to services running in the virtual device server to multiplex the usage of physical devices. The LynxSecure Virtualization Layer can emulate the following devices:

- Video Adapters
- Gigabit Ethernet Controllers
- AHCI/IDE Storage Controllers
- Audio Controllers
- USB Controllers
- PS/2 Keyboard, Mouse

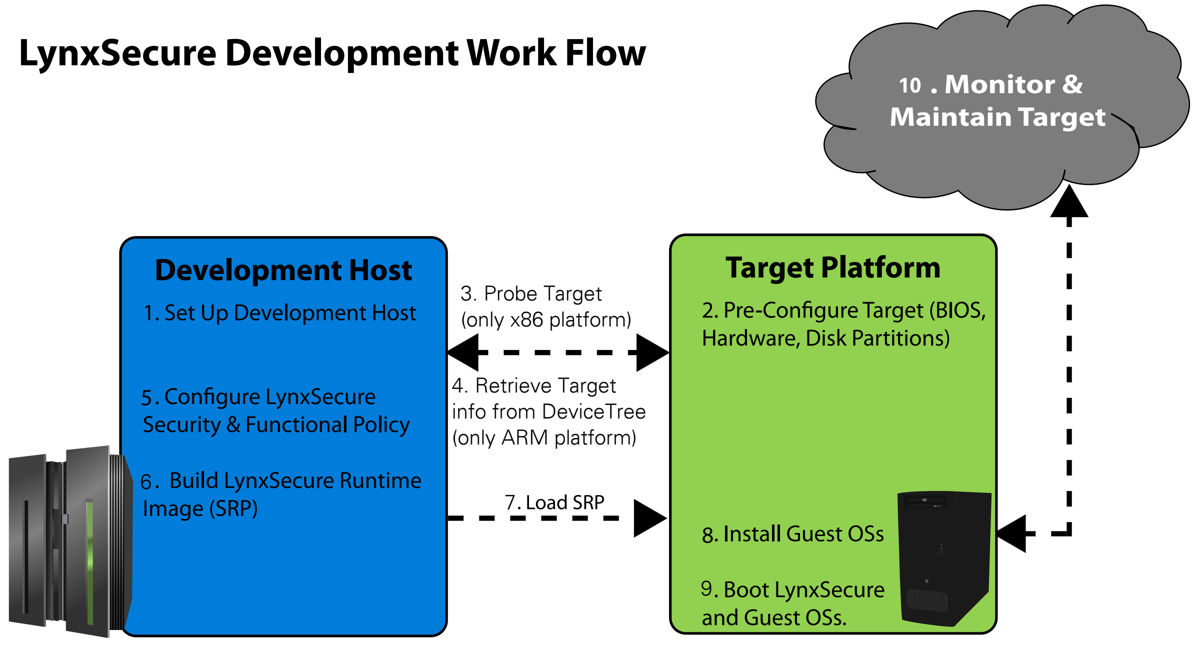
Hypercall API

The Hypercall API is a privileged interface from a Guest Operating System or LSA to LynxSecure for the purpose of providing privileged routines such as, inter-guest communication, Guest Operating System management and auditing. The Hypercall API plays a critical role in developing advanced solutions. See the *LynxSecure API Guide* for additional details on the Hypercall API.

CHAPTER 4 *LynxSecure Development Workflow*

This chapter illustrates standard start-to-finish workflow of using the LynxSecure® Development System.

Figure 4-1: LynxSecure Development Workflow


LynxSecure Development Work Flow
10 . Monitor & Maintain Target
Development Host
1. Set Up Development Host
5. Configure LynxSecure Security & Functional Policy
6. Build LynxSecure Runtime Image (SRP)
3. Probe Target (only x86 platform)
4. Retrieve Target info from DeviceTree (only ARM platform)
7. Load SRP
Target Platform
2. Pre-Configure Target (BIOS, Hardware, Disk Partitions)
8. Install Guest OSs
9. Boot LynxSecure and Guest OSs.

Installation, Configuration, Deployment, and Maintenance Outline

The following steps outline the process for installing, configuring, deploying, and maintaining the LynxSecure Development System:

1. Set up the Development Host

   Install the LynxSecure development tools on the host development platform. The development tools run on a CentOS™ 7.3 or later operating system. The development tools also rely on several services that must be configured including a license server, a DHCP server, and a network boot server. To get started quickly, it is recommended that the host and the target be connected via Ethernet. The tools and services can be installed on a pre-existing customer owned installation of CentOS 7.3 or later, or can optionally be pre-installed with the purchase of a development host platform.

2. Set up the Target System

Target systems running LynxSecure must be pre-configured to enable required features in the system firmware. For example, this step may include enabling VT-d support on x86 platforms.

3. Probe the Target (only applicable for a x86 target)

   Once the target is pre-configured, users must then run a probe tool from the development host. This tool captures the required target hardware and configuration profile needed to start the configuration process.

4. Retrieve target information using a Device Tree (only applicable for a Arm target)

   Once the target is pre-configured, users must then retrieve the target information using a command line tool. This tool captures the required target hardware and configuration profile needed to start the configuration process.

5. Configure LynxSecure

   Next, a system configuration is generated using command line tools. This configuration is stored as XML and fully defines the run time configuration of the system while executing LynxSecure. This XML configuration is called the Human-readable Configuration Vector, and is often referred to as the HCV.

6. Build the LynxSecure Runtime Image

   Using the HCV as an input, the configuration compiler generates the System Runtime Package. This binary contains the hypervisor along with any configured guest resources. The System Runtime Package is generally referred to as the SRP.

7. Load the SRP (LynxSecure System Runtime Package)

   Once the SRP bootable image is generated by the configuration compiler, the bootable image must then be loaded onto the target platform. The SRP can be booted from many media formats including a Local Hard Disk, Flash Media, USB Media, or the network.

8. Install the Guest Operating System(s)

   Now that the SRP is running on the target system, the guest operating systems can be configured and installed. While LynxSecure provides flexibility in this area, it is recommended that guest operating systems are installed from within a running instance of LynxSecure. This will ensure that all of the guest's resources are securely separated, and that no unauthorized resources will be configured during installation.

9. Boot LynxSecure & Guest Operating System(s)

   Once the guest operating systems have been installed, LynxSecure must be fully rebooted by safely shutting down each guest and resetting the whole system.

10. Monitor and Maintain the Target

    Using the development tools, users have the option to connect to reference monitor agents included in the Virtual Device Server or to custom developed maintenance, security, and reliability software that may either be embedded alongside the guest operating system as autonomous components, or within the management guest Virtual Device Server.