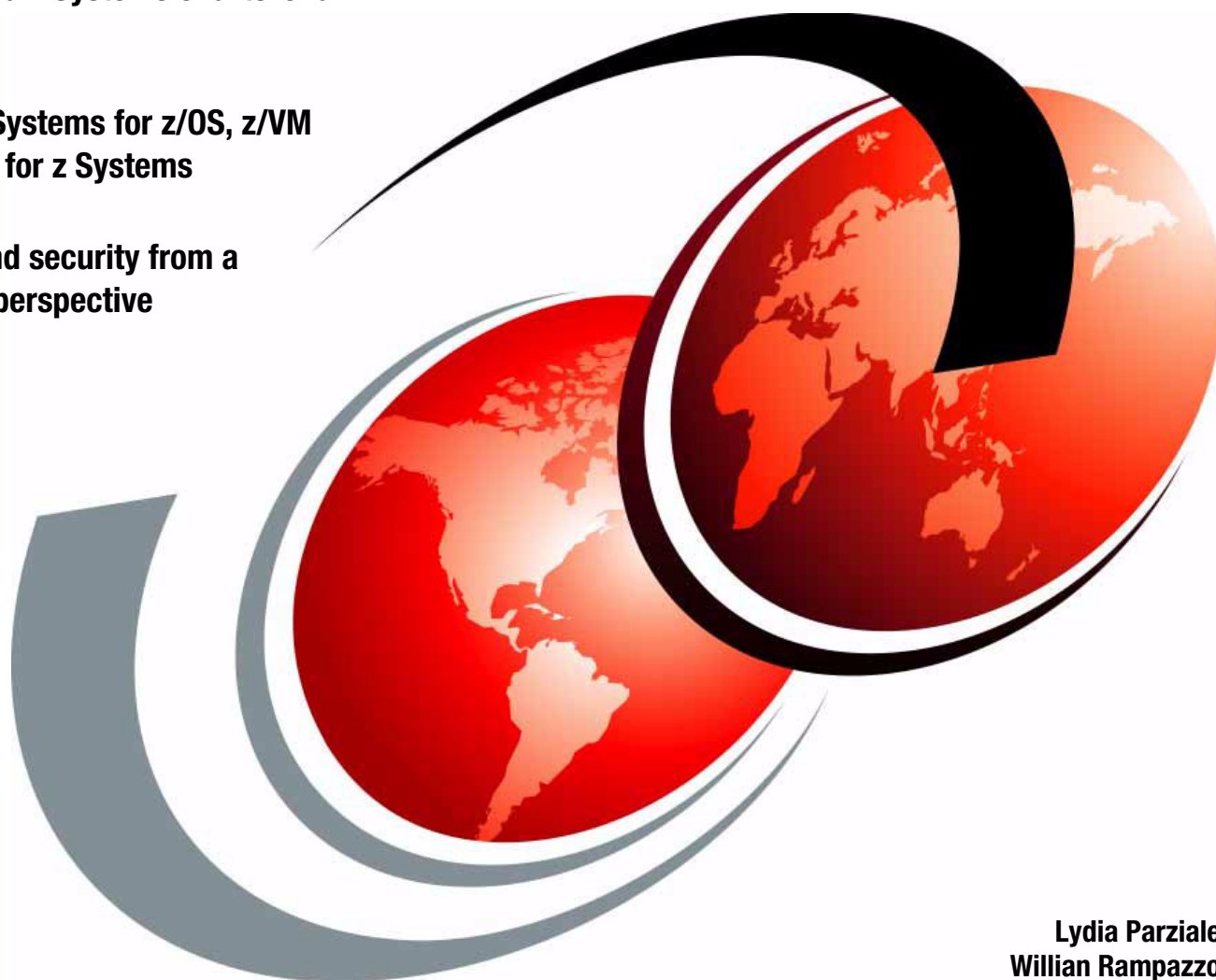


# End to End Security with z Systems

Understand z Systems end-to-end security

Secure z Systems for z/OS, z/VM and Linux for z Systems

Understand security from a use case perspective



Lydia Parziale  
William Rampazzo





International Technical Support Organization

**End to End Security with z Systems**

March 2015

**Note:** Before using this information and the product it supports, read the information in “Notices” on page v.

**First Edition (March 2015)**

This document was created or updated on March 30, 2015.

**© Copyright International Business Machines Corporation 2015. All rights reserved.**

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Notices</b> .....	v
Trademarks .....	vi
<b>Preface</b> .....	1
Authors .....	1
Now you can become a published author, too! .....	1
Comments welcome .....	2
Stay connected to IBM Redbooks .....	2
<b>Chapter 1. Introduction</b> .....	3
1.1 Systems of Record and Engagement .....	4
1.1.1 Systems of Record .....	4
1.1.2 Systems of Engagement .....	4
1.2 The security perspective .....	4
1.2.1 Information security principles .....	5
1.2.2 Security and z Systems features .....	5
1.2.3 Putting it all together .....	7
<b>Chapter 2. Security Overview</b> .....	9
2.1 Defense in depth .....	11
2.1.1 Protecting the physical IT infrastructure .....	11
2.1.2 Protecting the servers .....	12
2.1.3 When data physically leaves the data center .....	12
2.2 Inside the bits protection .....	13
2.2.1 Operating System Security - The Hypervisor .....	13
2.2.2 Operating System Security - the guest .....	14
2.2.3 Application security, protecting the database server .....	14
2.2.4 Application security, protecting the application server .....	15
2.2.5 Communication security .....	16
<b>Chapter 3. Use case</b> .....	17
3.1 Use case description .....	18
3.1.1 General architecture and transaction flow .....	18
3.1.2 Mobile device management .....	20
3.1.3 Securing the network .....	21
3.2 Securing the enterprise applications .....	25
3.3 Operating system security - z/VM .....	25
3.3.1 Virtual machine security .....	26
3.3.2 Hypervisor security .....	27
3.4 Data security .....	32
3.5 Operating system security - Linux on IBM z Systems .....	33



# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.


## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

# Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

CICS®	IMS™	WebSphere®
DB2®	InfoSphere®	z/Architecture®
DirMaint™	RACF®	z/OS®
DS8000®	Redbooks®	z/VM®
Guardium®	Redpaper™	z/VSE®
HiperSockets™	Redbooks (logo)  ®	zSecure™
IBM®	System z®	

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, or service names may be trademarks or service marks of others.



# Preface

This IBM® Redpaper™ provides a broad understanding of the components necessary to secure your IBM z Systems environment. It provides an end-to-end architectural reference document for a use case that employs both mobile and analytics. It also provides an end to end explanation of security on z Systems from the systems of record through the systems of engagement. Security is described in terms of transactions, covering what happens after a transaction hits the system of engagement and what needs to be in place from that moment forward.

The audience for this paper is IT architects and those planning to use z Systems for their mobile and analytics environments.

## Authors

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

**Lydia Parziale** is a Project Leader for the ITSO team in Poughkeepsie, New York, with domestic and international experience in technology management including software development, project leadership, and strategic planning. Her areas of expertise include business development and database management technologies. Lydia is a certified PMP and an IBM Certified IT Specialist with an MBA in Technology Management, and has been employed by IBM for over 25 years in various technology areas.

**Willian Rampazzo** is a Software Engineer at Linux Technology Center (LTC) in IBM and an Assistant Professor for a Computer Science bachelor course in Brazil. He has a Bachelor's degree in Computer Science and an MBA in Information Technology Security Management. He has been working with z Systems for 12 years at IBM, and has worked as a z/VM® system programmer and a Linux on z Systems administrator for IBM Internal Account. He is now focused on development for Linux on z Systems at LTC

Special thanks to Nigel Williams from the IBM Client Center, Montpellier, France for assisting in creating the use case discussed in this book.

## Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks® publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- ▶ Send your comments in an email to:

[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400

## Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- ▶ Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



# Introduction

This chapter explains the importance of end to end security on z Systems as it relates to both the systems of record and the systems of engagement. It also covers is the difference between security and integrity and the importance of each.

This chapter includes the following sections:

- ▶ Systems of Record and Engagement
- ▶ The security perspective

## 1.1 Systems of Record and Engagement

This section describes the unique qualities of the systems of record and systems of engagement.

### 1.1.1 Systems of Record

The Systems of Record are largely geared towards passively providing information to a company's workers. "Systems of Record" are the enterprise resource planning (ERP)-type systems companies rely on to run their businesses today (financial, order processing, customer relationship management, human resources, and so on). They must be "correct" and "integrated" so that all data is consistent. And they are traditionally designed for people who have no choice but to use them. They passively provide information to the enterprise and its business partners, often through a firewall.

A large percentage of the data and transactions, as well as massive data warehouses for business analytics, originate or reside on IBM z Systems.

Systems of Record on z Systems are mainly on-premises, core transactions based in z Systems that benefit from the security that is designed *into* z Systems.

### 1.1.2 Systems of Engagement

Geoffrey Moore, speaker, advisor, and author of many business-evolution-related books, introduced the concept of Systems of Engagement as the Future of IT and the transition in IT investment focus from Systems of Record to Systems of Engagement.

Systems of Engagement revolves around the transition from current enterprise systems designed around discrete pieces of information (the records) to systems that are more decentralized, incorporate technologies that encourage peer interactions (such as Facebook and Twitter), and often use cloud technologies to provide the capabilities to enable those interactions (such as mobile applications). These systems often must provide 24x7 indicators of the health of a business (analytics) as well.

## 1.2 The security perspective

Systems of Engagement is an evolution of IT into a more collaborative based environment, and hence, security technologies must evolve as well. When moving from Systems of Record to Systems of Engagement, the concern becomes about not only giving employees and partners access to the enterprise network without compromising it, but also controlling and managing consumer access to data. Of course, there are firewalls for the consumers, but now companies also want to create personalized, rich media experiences at the edge, through real-time processing. All this creates a new raft of architecture challenges.

This section introduces this IBM Redpaper's prime topic, which is the end-to-end security challenges from a z Systems perspective in the context of information security principles.

## 1.2.1 Information security principles

Regardless of the form that data might take, information security is the practice of protecting and defending information from unauthorized access and use, disclosure, disruption, or modification. All of these form the triad of basic information security:

Confidentiality	The set of rules or a promise that limits access or places restrictions on information. It is basically equivalent to privacy. Confidentiality involves protecting data from unauthorized access or disclosure.
Integrity	The assurance that the data being accessed or read has not been tampered with or been altered or damaged since the last authorized access. Integrity involves maintaining and assuring the accuracy and consistency of data through its lifecycle.
Availability	Information is available when it is needed. Availability is synonymous with uptime when discussing hardware. When considered in the larger picture, uptime is not just a function of hardware, but also of software stability and resilience to disaster or attack. Availability is about resilience, business continuity, and disaster recovery. You must ensure that backup information and systems are in place for recovery purposes

These three elements of the triad are considered to be the most crucial components of security. The purpose of information security is to preserve these three elements.

## 1.2.2 Security and z Systems features

This section provides a high-level overview of the z Systems features and benefits, and how the various operating systems and applications are able to use the built-in z Systems security features.

### Hardware

The IBM z13 as the newest addition to the mainframe has a rich history of delivering a secure infrastructure. It is built on a set of hardware security capabilities that include multi-state operation modes, storage key isolation, high-speed standards-based encryption, logical partitions, and many other features and benefits. The following are some of these features and benefits that enhance the security of the IBM z13:

- ▶ Cryptographic cards
  - Crypto Express5S is an accelerator feature capable of intelligent encryption of sensitive data run faster than previous generation Crypto Express4S, with less disk requirement and centralized key management
- ▶ CP Assist for Cryptographic Functions (CPACF)
  - Cryptographic processor unit available on every core, enabled as a no charge feature
- ▶ EAL5+ (Common Criteria Evaluation Assistance Level 5+) certified, a regulatory certification for logical partitions (LPARs) verifying separation of partitions to improve security
- ▶ Elliptic Curve Cryptography (EEC)
  - Cryptographic capabilities that are designed to provide public key support for constrained digital environments
- ▶ Trusted Key Entry (TKE)
  - A feature that is a means for ensuring secure creation and management of key material and for managing the crypto adapters on the host

## Operating system

IBM z Systems servers support multiple operating systems. This section provides an overview of the security features available for the operating systems IBM z/OS®, z/VM, and Linux on z Systems.

### ***z/OS***

The security of z/OS is centralized on the System Authorization Facility, which can provide its own security services, but is more likely to route requests for security services to another security manager such as the IBM Resource Access Control Facility (RACF®).

Building on the hardware foundation, the z/OS operating system provides software security capabilities such as RACF, and the Integrated Cryptographic Service Facility (ISCF). Technologies such as Secure Sockets Layer (SSL), Kerberos V5, Public Key Infrastructure, multilevel security, and exploitation of IBM mainframe cryptographic features are all available in z/OS.

The following are additional features and benefits:

- ▶ RACF and IBM Distributed Identity Data (IDID) provides discrete, end to end authentication, transaction auditing, and identity mapping
- ▶ Cryptography options supporting advanced encryption processing
- ▶ PKI services centrally manage certificates
- ▶ High-level security connection to backend applications through IBM HiperSockets™

### ***z/VM***

IBM z/VM is the z Systems virtualization platform. It supports more virtual servers than any other in a single footprint. z/VM provides isolation and protection of virtual machines from each other, and between virtual machines and the system overall. These functions are provided by the z/VM Control Program (CP) and supported by features of the IBM z/Architecture® and z Systems hardware. Although the core capability of security and integrity is provided by CP, without an external security manager (ESM) such as RACF, the management of this capability is quite basic.

z/VM provides the following features and benefits in support of a more secure environment, among others:

- ▶ The z/VM hypervisor
  - The z/VM hypervisor, which refers to a system that virtualizes the real hardware environment, is designed to help extend the business value of mainframe technology while providing availability, security, and operational ease. The hypervisor, also called a virtual machine manager, is a program that allows multiple operating systems to share a single hardware host. Each operating system appears to have the host's processor, memory, and other resources all to itself. However, the hypervisor is actually controlling the host processor and resources, allocating what is needed to each operating system in turn and making sure that the guest operating systems (called virtual machines) cannot disrupt each other.
- ▶ Cryptography options supporting advanced encryption processing
  - Supports Crypto Express5S feature
  - Can be configured as coprocessors for secure key transactions
  - Can be configured as accelerators for SSL communications

### ***Linux on z Systems***

In addition to the security provided by the z/VM operating system, Linux on z Systems guests are also provided with a Pluggable Authentication Module (PAM) LDAP function to improve

security and to reduce administrative tasks. With the PAM function, you can avoid defining users under a z Systems Linux in every Linux server and can have a unique user ID and password on z/VM, z/OS, and Linux.

For more information about security for Linux on z Systems, see *Security for Linux on System z*, SG24-7728.

## Applications

z Systems is the only commercial operating system that has achieved EAL 5+ certification. This certification means that although different workloads are running on the same hardware, they are protected when running in separate partitions. One LPAR cannot reach across boundaries into the next LPAR and compromise its security. The LPARs are allocated their own resources, and are secure and separate environments.

Additionally, the CPACF has been redesigned to allow for the handling of higher volumes of transactions. Many applications use these CPACF benefits, among others:

- ▶ IBM InfoSphere® Guardium® Data Encryption for DB2® and IMS™ Databases
- ▶ DB2 built-in encryption
- ▶ z/OS Communication Server: IPsec/IKE/AT-TLS
- ▶ z/OS System SSL
- ▶ z/OS Network Authentication Service (Kerberos)
- ▶ IBM Encryption Facility for z/OS DFSMSdss encryption feature
- ▶ IBM Encryption Facility for z/OS
- ▶ z/OS Java SDK
- ▶ Linux on IBM System z®; kernel, openssl, openCryptoki, GSKIT

## Storage

The IBM DS8000® disk storage series offers a self-encrypting disk that uses IBM Full Disk Encryption (FDE) disks and flexible key manager software. The DS8000 encryption secures data at rest and offers a simple, cost-effective solution for securely erasing any disk drive that is being retired or repurposed (cryptographic erasure).

For more information about how even the z13 storage is secured, see *IBM DS8870 Disk Encryption*, REDP-4500.

## Communications

Securing the communication is important to keep the integrity of a request and the integrity of the data going through the communication. The following are some ways that the IBM z Systems secure communications:

- ▶ CPACF provides communication encryption (to applications such as the IBM HTTP Server).
- ▶ At the operating system level, the hypervisor provides isolation and integrity of virtual servers so that bad behavior by one cannot compromise the execution or data that are associated with another.

### 1.2.3 Putting it all together

IBM z Systems, with its enhanced security, brings your systems of record together into your systems of engagement in a secure environment that can support the volume of transactions that mainframe users are accustomed to.

As an example, in today's mobile era, over 10 billion devices are accessing information. Enterprises are challenged with integrating new mobile services with existing organizational

processes, without sacrificing the client's experience. Approximately 70% of all enterprise transactions involve IBM z Systems. The new IBM z13 with its enhanced data processing capabilities can play an important role by providing the secure and stable base that you need to extend your existing enterprise data and transactions to mobile users. Figure 1-1 shows how, in a typical environment, access to applications and interaction with the systems is achieved from mobile devices.

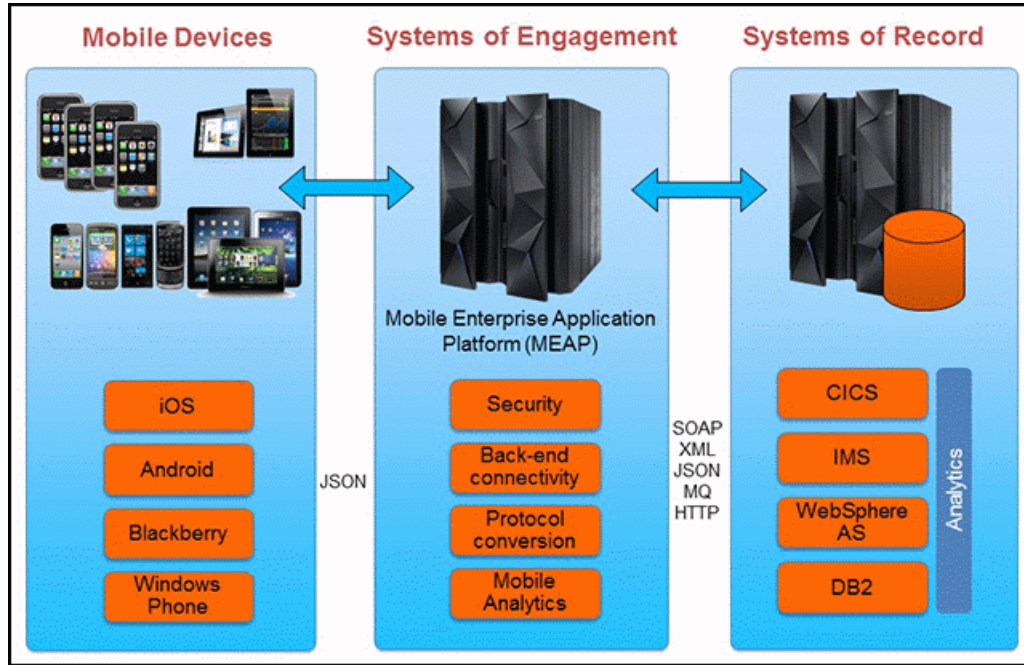


Figure 1-1 From systems of record to systems of engagement to your mobile devices

The IBM z13 maintains levels of security at each step of the way, from systems of record to systems of engagement to the mobile devices.



## Security Overview

The world is becoming more digitized and interconnected, which open the door to emerging threats, leaks, and attacks. The average cost of a security breach is \$5.8 million US dollars (USD)! Analytics, mobile, social, and cloud computing all have one thing in common: They need a platform that has a deeply integrated security stack, as shown in Figure 2-1.



*Figure 2-1 Analytics, mobile, social, and cloud computing require a deeply integrated security stack*

When talking about security, it must be applied to the environment as a whole, and not to a single machine, operating system, or application.

In general, security must be applied to the physical location of an organization, on notebooks or desktops employees use to perform their daily work. It must be applied on data centers, servers, and network. Think about security as an onion, where layers involving the core can be considered as levels that need to be reached combined with the security access.

For example, to reach core data, you must access a data center, then access the server, and then access the console of that server using an administrator user ID. Access must be granted at each level, from physical to logical.

This chapter provides an overview of this kind of layered security, used by most companies, which is known as defense in depth.

This chapter includes the following sections:

- ▶ Defense in depth
- ▶ Inside the bits protection

## 2.1 Defense in depth

The first steps to make an organization less exposed to the threats and vulnerabilities of a business environment are to identify and employ the best security requirements for the business. Based on that, create the policies that guide the overall configuration of the IT infrastructure.

To grant access to different layers of a defense in depth model, an established information security policy must be in place that defines the roles and the rules for controlling access to data.

For most companies, the most valuable asset is the *information* and to secure it, the first step is to create an information security policy. The basic purpose of an information security policy is to protect the data. A constant challenge is to make information always available and at the same time always secure.

The start point is setting rules for users based on their expected behavior. For example, system administrators will have different privileges than management or security personnel. As a default, roles must always be defined with the least access needed. This definition helps to keep a strict control of who is accessing the data.

A complete information security policy also covers the physical security of desktops, notebooks, servers, and data centers.

When an information security policy is in place, the role of security personnel or team must have access to monitor, probe, and investigate any suspect activity on a system, network, or data center. The security role must also be explained, in detail, in the information security policy as well as the consequences for any violation.

An information security policy will not ensure total security alone. It is a framework that contains preferred practices, rules, and roles that the employees will follow. It helps minimize the risk of a violation and also helps tracking compliance with regulations and legislation.

To make the information security policy valid, all employees in an organization must have access to it, must read and understand it, and usually physically or electronically sign it, agreeing that they have read it. This turns staff into participants in the effort to secure information assets.

### 2.1.1 Protecting the physical IT infrastructure

Following the analogy of the onion, the first layer of an IT infrastructure that must be protected is the physical access to the data center. Typically, only maintenance people, like hardware support personnel, network support personnel, facility support personnel, and a restricted number of system administrators need to have access to data centers. However, that is a decision that should be made while developing the information security policy for the organization.

The roles of those who have access to a data center can vary from organization to organization. There are cases where access to the data center is not part of the role of some people, but they will occasionally need to access the data center. This is also acceptable, must be stated in the information security policy, and must be controlled for audit purpose. Access from people who do not maintain the data center can cause unnecessary problems such as mistakenly turning a server turn off or unplugging a network cable.

Some solutions can be used to control the physical access of a data center. It mainly depends on how the organization wants to protect the data center and how much the organization is willing to spend securing the data center.

Choosing what kind of security access will be in place in the data center is an important decision considering information is one of the most valuable assets of an organization. Some options to accomplish data center security access are using locks, biometrics, guards, identification badges, or any other way that prevents unauthorized access from people and grant access just to permitted people.

As an example, in an organization that has a data center accessed by a few people, a lock is typically enough. In a data center that does not have a card reader mechanism, a guard can verify employees identification badges. More than one method to grant access can also be used to make the data center security stronger. As an example, a guard can verify employees identification badges and make sure that they are the real owners of it and a card reader can grant or deny the access to the data center.

### **2.1.2 Protecting the servers**

After access to the data center is in place, it is time to establish the security of the servers and consoles in the data center. Not everyone who accesses the data center must have access to all server consoles in the data center.

A preferred practice is that all server consoles have a way to request authentication to be unlocked for use. It can be an authorization method to unlock the window, if just an administrator user ID is used at the console. It can also be a user ID authorization and authentication method to make sure that the person accessing the console has the correct privileges to the server, considering the operating system has a separation of privilege classes for user ID.

### **2.1.3 When data physically leaves the data center**

Information that physically leaves a data center must also be protected.

There are basically two ways the information physically leaves a data center. First is when moving machines to another building or data center. Second is when moving backup media offsite.

A good backup policy, known as the three-two-one rule, keeps three copies of the files: The original and two more backup copies. It stores the backup copies on two different media types, such as hard disk drives and magnetic tapes, and keeps one backup copy offsite. The offsite copy is kept in case of fire or flood in the data center building in which the other copies of the data are damaged.

In keeping with a good backup policy, media therefore must leave the data center with data backup. This media needs to be protected to avoid unauthorized access to the data. A good approach is to encrypt the data on this external media. This often prevents unauthorized access to the content of the media and protect the data on the media.

## 2.2 Inside the bits protection

Going further along with the layered analogy of the onion, with the physical security of the hardware in place, now it is time to secure the logical information, the bits. At this point, there are different layers that need to have security in place.

Starting from the server side, the operating system, the application, the database, and the communication between the client and the server need to be secure. Securing the communication is important to ensure the integrity of a request and the integrity of the data going through the communication.

### 2.2.1 Operating System Security - The Hypervisor

Hypervisor security will not differ from the security of any other operating system on a server. However, the virtual infrastructure relies on the security of the hypervisor, so protecting the hypervisor (in either IBM z/VM or KVM) typically prevents attempts to breach the security of the operating system and inadvertently compromising the integrity of the operating system and data.

Although each guest can have its own security configuration and faces threats particular to it, it is essential to protect the hypervisor itself as an equally important part of an overall end-to-end security policy. This is because actions like create, change, and remove virtual machines are performed at the hypervisor. Access to the virtualization management system should be restricted to authorized administrators only.

The work to maintain the hypervisor is part of the system administrator role. Depending on the infrastructure size of the organization, the roles of hypervisor system administrator and guest system administrator can be separated.

The hypervisor system administrator must ensure that the hypervisor is secure. A best practice is to always keep the hypervisor up to date with corrections. Installing the corrections as soon as they are released decreases the time frame the vulnerability can be exploited. Use of centralized patch management solutions helps with the increase in number of hypervisors managed on an IT infrastructure.

Another practice that helps avoid undesirable problems is to disconnect or undefine unused physical hardware from the hypervisor and disable all unnecessary services on the hypervisor operating systems. Doing so decreases the number of vectors an attacker can use to get access, attempt to breach, or compromise the hypervisor, and through it the operating system.

Besides operating system setup and customization for security, monitoring the hypervisor for signs of compromise helps you to promptly respond to a threat. Use monitoring tools to help monitor the hypervisor and also look at the hypervisor logs for suspicious activities, both of which make the work of the hypervisor system administrator easier.

During the infrastructure planning, a practice that helps secure the hypervisor is to separate the hypervisor network and its guests network. If possible, restrict the hypervisor network to administrative teams using a dedicated management network, or encrypt the management network communications to the hypervisor operating system.

## 2.2.2 Operating System Security - the guest

A guest operating system running on a virtualized environment does not differ from an operating system running on real hardware in terms of security. The same security considerations that apply to operating systems running on real hardware also apply to guest operating systems running in a virtualized environment.

In this case, all recommended practices for managing operating systems running on real hardware such as update management, log management, authentication, remote access, and user ID management, must be followed. Nevertheless, for the guest operating system some additional security actions should be taken.

The first action to secure guest operating systems is to disconnect any unused virtualized hardware. An example of unused virtualized hardware that can expose the guest operating system is a network adapter or a removable media drive. Removing the unused virtualized hardware is important to reduce the vectors that an attacker can use to attempt to breach, get access to, or compromise the guest operating system.

Another action used to secure your guests concerns shared disks among the guests. If one of the guests is compromised, the shared disk can be used to compromise the other guest systems that access the same shared disk. If the organization includes rules for shared network-attached storage access in the information security policy, these rules should apply to the shared disks on a virtualized environment as well.

If the guest systems are not related to each other, for example, if they are used by different customers, separate the authentication methods for them to ensure that users are not authorized to authenticate on guest systems they are not supposed to use. A preferred solution is to separate customer guest systems on different networks altogether.

## 2.2.3 Application security, protecting the database server

Basic database security measures like authentication, authorization, and access control are no longer enough because of the growing number of sophisticated attacks. It is not unusual for a database administrator to discover a security breach long after it happens.

Before implementing database security, it is important to plan and know why the database must be protected. The following are some examples of securing the organization databases:

- ▶ Understand applicable regulatory compliance standards such as from the European Union (EU), Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry (PCI), and Sarbanes-Oxley (SOX).
- ▶ Know what data must be protected, such as credit card information, employee information, customers information, and projects information.
- ▶ Make a list of all databases within the organization, including those used for testing and development.
- ▶ Classify all the listed databases as highly sensitive, sensitive, and nonsensitive.
- ▶ Establish the policies that must be applied to each category of database. Depending on the category and as needed, implement advanced security measures such as firewall, encryption, masking, auditing, and monitoring.
- ▶ Integrate the database policies with the organization information security policy.

These actions can be defined using the three main pillars for enterprise database security:

- ▶ Foundation, the starting point for a database implementation strategy:
  - Discovery and classification of databases
  - Authentication, Authorization, and Access Control
  - Patch management
- ▶ Preventive, make sure the database is healthy:
  - Change management
  - Data masking
  - Network communication and data at rest encryption
- ▶ Detection, always be aware of what is happening:
  - Vulnerability assessment
  - Security monitoring
  - Database auditing

## 2.2.4 Application security, protecting the application server

An application server is used to host business logic and data access services. It can be part of the graphical user interface front end, but generally have it separated between the front end and the database to provide even more security. There are four basic approaches to attacking an application server<sup>1</sup>:

- ▶ Network-based attacks: These attacks rely on low-level access to network packets and attempt to harm the system by altering this traffic or discovering information from these packets.
- ▶ Machine-based attacks: In this case, the intruder has access to a system on which IBM WebSphere® Application Server is running. Your goal is to limit the ability to damage the configuration or to see things that shouldn't be seen.
- ▶ Application-based external attacks: In this scenario, an intruder uses application-level protocols (HTTP, IIOP, JMX, web services, and so on) to access the application, perhaps by using a web browser or some other client type. The intruder uses this access in an attempt to circumvent the normal use of the application usage and do inappropriate things. The key is that the attack is run using well-defined APIs and protocols. The intruder is not necessarily outside the company, but rather is running code from outside of the application itself. These types of attacks are the most dangerous because they usually require the least skill and can be done from a great distance if IP connectivity is available.
- ▶ Application-based internal attacks (also known as application isolation): In this case, you are concerned with the danger of a rogue application. In this scenario, multiple applications share WebSphere Application Server infrastructure, and you do not completely trust each application.

The Java EE specification and WebSphere Application Server provide a powerful infrastructure for implementing secure systems. Secure Sockets Layer/ Transport Layer Security (SSL/TLS, hereafter referred together as SSL) is a key component of the WebSphere Application Server security architecture, used extensively for securing communication. SSL is used to protect HTTP traffic, IIOP traffic, LDAP traffic, MQ traffic, JDBC traffic, messaging traffic over the SIBus between WebSphere messaging engines, and J2C and SOAP traffic. SSL requires the use of public/private key pairs, and, in the case of WebSphere Application Server, these keys are stored in keystores.

<sup>1</sup> [http://www.ibm.com/developerworks/websphere/techjournal/1210\\_lansche/1210\\_lansche.html](http://www.ibm.com/developerworks/websphere/techjournal/1210_lansche/1210_lansche.html)

The combined benefits of IBM Java 8 and z13 features (including single-instruction multiple-data (SIMD) vector engine, simultaneous multithreading (SMT), and improved CP Assist for Cryptographic Function (CPACF)) provide up to 2X improvement in throughput-per-core for security-enabled applications and up to 50% improvement for other generic applications.

Application serving with SSL uses the new Java 8 Clear Key CPACF and SIMD vector instructions for string manipulation.

## 2.2.5 Communication security

The previous sections described how all layers are individually secured to keep data integrity, data confidentiality, and data availability at the source.

Security in individual layers might be enough to keep the data integrity, confidentiality, and availability at the destination, but it is important to secure the data while it is in transit, during communication.

Using the z13 cryptographic hardware, you gain security from using the CPACF and Crypto Express5S through in-kernel cryptography APIs and, for Linux on z Systems, the libica cryptographic functions library. Using these features provides these benefits:

- ▶ File system encryption
- ▶ Communication encryption (to the applications such as IBM HTTP Server)
- ▶ System security by providing advanced cryptographic functions

Some solutions can be implemented at the client side, but the organization cannot rely on client-side only security. Users can forget to update their security software or security operating system updates, can unconsciously install malware on their devices that prevents the execution of the security software, or the users just do not install the security software.

What the organization can do is make sure the communication between the client and the server is encrypted with a secure cryptographic protocol. New vulnerabilities are often discovered on cryptographic protocols, cipher algorithms, and protocol implementation, so the security team need to be up to date about what is currently secure to be used and new vulnerabilities that need to be mitigated as soon as reported.

The IT infrastructure inside the organization is the responsibility of the organization, so all means to avoid security breaches are valid to protect the information. A well planned network infrastructure also helps secure the data communication. The first point of contact with the Internet should be the network security system. It controls the incoming and outgoing traffic to the organization's network based on the application set of rules.

Separating the network into layers helps protect the information. Therefore, during network infrastructure planning, consider at least a layer for a DMZ, a layer for the web servers, a layer for the application servers, and a layer for database servers. This is not a rule and can be structured in different ways, but layering the network is important and must be considered when planning the network infrastructure.

Installing intrusion detection systems assists in monitoring for attacks and helps parse audit logs that can use a large amount of storage and have a huge amount of information that a human will not be able to read and find a pattern for an attack at the same time it is happening. Intrusion detection systems help system and network administrators detect attacks and alert them about it while it shows the techniques in use to exploit possible breaches.





## Use case

To keep the integrity of data in your system, secure all devices that serve as a path or store the information such as a mobile device and web browser used as an interface to the system. Also secure the network infrastructure that connects the client to the server, the hypervisor, and guest operating system that holds the application server and the database server, and also protects the backup media.

This chapter introduces a typical business scenario of a fictional company securing both a website and a mobile application. It describes the business and technical requirements for end-to-end security using z Systems from the systems of record through the systems of engagement on out to your mobile application.

This chapter includes the following sections:

- ▶ Use case description
- ▶ Securing the enterprise applications
- ▶ Operating system security - z/VM
- ▶ Data security
- ▶ Operating system security - Linux on IBM z Systems

## 3.1 Use case description

Mainframe transaction processing systems (IBM CICS®) and data management products (DB2, IMS, and VSAM) are involved with most of the data stored on a mainframe. Recall from Figure 1-1 on page 8, how you would have an application server in the systems of record that can be used to serve data from the systems of record to the web as well as from the systems of record to the systems of engagement and later to your mobile applications.

This scenario describes a general insurance application, GENAPP. GENAPP is a mobile application where a customer can perform the following functions:

- ▶ View insurance policies
- ▶ Apply for new Home, Car, or Endowment policy
- ▶ Start an insurance claim

The mobile application makes calls to the insurance application (GENAPP) which is a CICS COBOL application that the insurance company uses to create and manage customer and insurance policy data. The data flows from the systems of record (CICS) into the systems of engagement.

For a more information about how to implement this use case, see Chapter 11 of *IBM System z in a Mobile World Providing Secure and Timely Mobile Access to the Mainframe*, SG24-8215.

### 3.1.1 General architecture and transaction flow

Figure 3-1 demonstrates the architecture and transaction flow of the GENAPP mobile application.

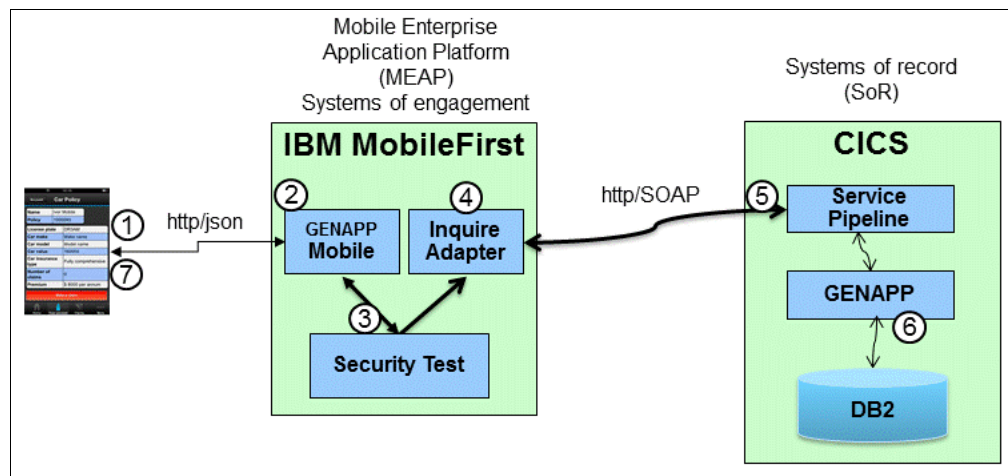


Figure 3-1 Insurance application architecture and transaction flow

The numbers in Figure 3-1 correspond to these steps:

1. The mobile application user logs in to the application.
2. IBM MobileFirst application receives request.
3. IBM MobileFirst application calls security test.
4. IBM MobileFirst adapter transforms mobile request and calls CICS to retrieve customer policies.

5. CICS web service framework converts the request to the channel interface of the GENAPP application.
6. GENAPP Cobol application processes the insurance policy request and queries the GENAPP database.
7. The policy data is returned to the mobile app.

In terms of security, IBM MobileFirst facilitates user authentication and integration into existing security systems. It provides an integration with back-end applications, systems, and services. Existing applications and database management systems can be accessed directly through common service oriented architectures and connectors that are provided by the IBM MobileFirst Platform.

Mobile security implementation requires a strategy for securing three levels (Figure 3-2):

- ▶ Mobile device management: Secure the endpoint device and its data, including the mobile application itself. In this case, GENAPP.
- ▶ The network or channel: Authenticate, encrypt, monitor and manage, control and block.
- ▶ The mobile application: Secure access to the enterprise applications and data.

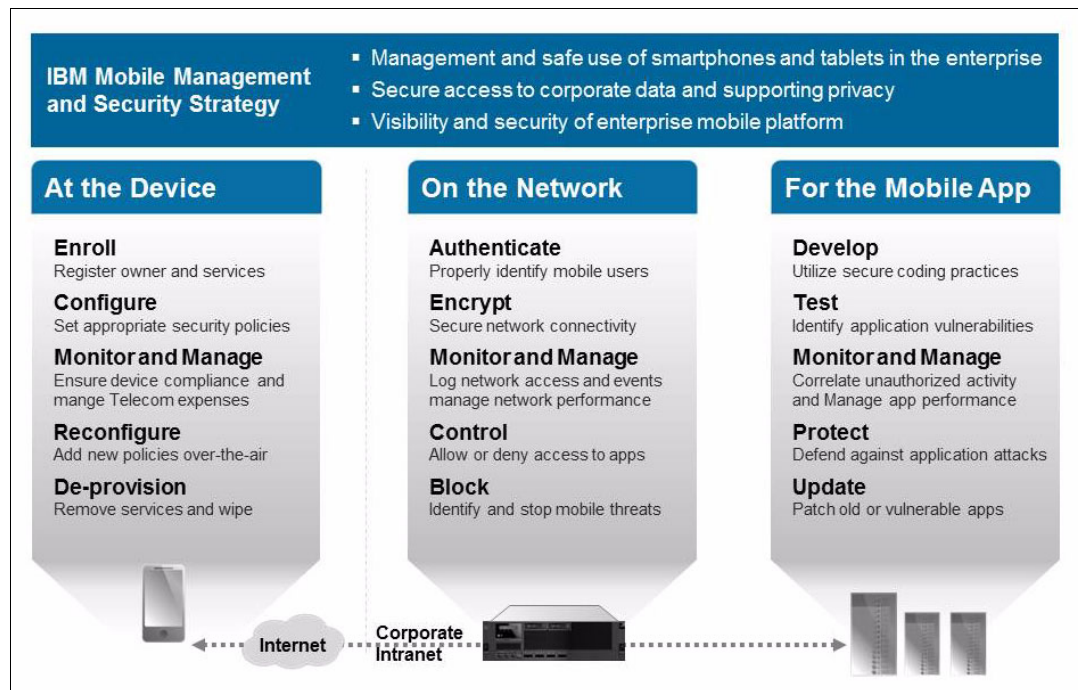


Figure 3-2 Securing three levels for mobile security

Figure 3-3 depicts how risk mitigation controls for mobile security need to be deployed at all three levels.

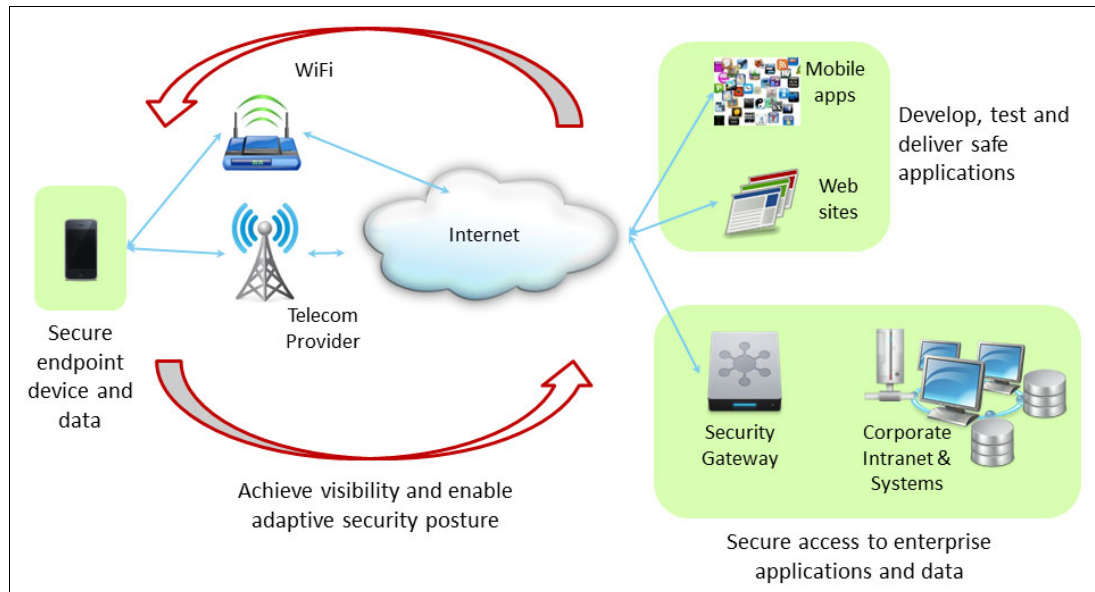


Figure 3-3 Mobile security architectural overview

### 3.1.2 Mobile device management

To manage devices and enforce corporate security policy, organizations use mobile device management (MDM) platforms. The MDM platform can also perform policy compliance assessments, device wipes, application management, and device lockdowns. Organizations that want to manage mobile devices typically require their employees to install an agent or a mobile application embedded with a management agent. Each user then must register and activate the device before it can be used for business. Given the management resources required, self-management capabilities can be offered to the employees to improve the responsiveness of the solution.

Mobile applications offer a host of possibilities for enterprises, but only if implemented correctly and securely. This section describes securing your mobile applications. MDM solutions help organizations manage multivendor mobile devices through a central console, and help to enforce these good practices, among others:

- ▶ Password for user authentication
- ▶ Automatic locking if the device is idle for a certain period
- ▶ Anti-virus software and signature updates
- ▶ Auto wipe or erasing of data after a certain number of failed password attempts
- ▶ Self-service to remotely lock, locate, or wipe sensitive enterprise data on a device if stolen or lost
- ▶ Securing sensitive enterprise data by using, for example, encryption, containerization, or virtualization

An emerging trend when deploying a mobile device management solution is to segregate the personal profile and the business profile on the mobile device, and to manage only the business profile. IBM MaaS360 is one comprehensive enterprise MDM platform that can provide end-to-end support for users, devices, emails, apps, documents and web access.

For more information about the various approaches to mobile device management, see *Securely Adopting Mobile Technology Innovations for Your Enterprise Using IBM Security Solutions*, REDP-4957, and *IBM System z in a Mobile World Providing Secure and Timely Mobile Access to the Mainframe*, SG24-8215.

### 3.1.3 Securing the network

When using enterprise applications, mobile users must have secure connections into the organization, and there must be adequate protection to keep the perimeter secure. A central location to manage and control access into the organization makes it easier to control and manage secure entry.

By implementing mobile systems of engagement on z Systems, with access to systems of record as z Systems back-end services and data, an entire network topology can be virtualized on the z Systems platform. The official certified high isolation level for z Systems and the virtual server that it can provision, enable the simplification of the entire network without an impact to the overall network security.

There are a number of advantages in flattening the network for any end to end application services on z Systems. This is particularly true for mobile applications due to the high traffic volumes.

This section describes the various ways of securing your z Systems network.

The heterogeneous environment on z Systems can be connected with the z Systems internal networks. There are three different types of z Systems internal networks between the systems of engagement and systems of record:

1. HiperSockets networks implemented in firmware

HiperSockets networks are within the physical z Systems server. You can define multiple independent HiperSockets networks in a single z Systems machine.

2. Shared Open Systems Adapter (OSA)

The z Systems network cards, called OSA, are used to build networks between servers and virtual images on z Systems and the enterprise networks.

3. z/VM virtualized networks with the VSWITCH function

The VSWITCH function in z/VM enables mapping of real networks into virtualized networks, which defines a secured network simplification and centralized management of a network topology. Multiple VSWITCH can be defined in a single z/VM environment that represent hubs in isolated networks in the same z/VM virtualization layer.

Network security is a wide subject to be explored. Security on every network path must be managed. It is possible to think about network security when connecting from the client software, a Telnet client for example, to a z/VM administrator user ID, or when connecting from client software, an SSH client for example, to an administrator user ID at a Linux on z Systems virtual machine running on z/VM, or even when connecting to the insurance system hosted on a Linux on z Systems virtual machine at a z/VM.

All of these types of network connections need to be secure to decrease the possibility of a breach in the IT infrastructure of this insurance company example.

During IT infrastructure planning, decide whether the network access to the hypervisor systems will be the same as the network access to guest systems. A preferred practice is to separate both networks, creating one for hypervisor management and one for virtual machine

management and the application access. If this is not possible, at least keeping an encrypted connection to the hypervisor systems is necessary.

## Network access to z/VM

Considering the z/VM network is in the same network at the guest systems, use of encrypted communication can be used to increase the security of the IT infrastructure for the insurance company. By default, Telnet 3270 session data flows unencrypted over the network, in clear text, meaning that anyone who dumps the network traffic is able to see what is happening between the Telnet client and the z/VM guest console.

Transport Layer Security (TLS), and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols to provide end-to-end encrypted communication. Digital certificates and trust hierarchies can be implemented to use encrypted communication.

When talking about SSL/TLS for z/VM, SSLSERV is a CMS service machine that provides encrypted communication to clients connecting to z/VM. Its code is preinstalled as part of standard z/VM installation and can be customized and enabled to provide SSL/TLS connections.

It is also possible to use z/VM FTP server for file transmission from a client software to the hypervisor and, by default, FTP server on z/VM do not accept any secure connection. However, it is possible to configure it to allow or require secure connections.

When using z/VM TCP/IP stack, different service virtual machines (SVMs) can be set up for different services, separating their functions related to specific protocol of support. As usual, the preferred practice is to enable just services that will be used and make sure that all security recommendation are followed.

The following are some examples of SVMs:

- ▶ TCP/IP service machine for basic connectivity with the hypervisor using Transmission Control Protocol / Internet Protocol (TCP/IP)
- ▶ FTPSERVE for File Transmission Protocol (FTP)
- ▶ LDAPSRV for Lightweight Directory Access Protocol (LDAP)
- ▶ MPROUTE for the communication route

Most of the TCP/IP stack SVMs can have security controlled by RACF/VM. This allows RACF/VM to process user ID connection authentication and authorization to the system and to resources, increasing the level of security on the system.

For more information about how to customize and enable the SSLSERV service system or configure FTP server to accept secure connections, see *Security for Linux on System z*, SG24-7728.

## Network access to Linux on z Systems

To connect to a Linux server using encrypted communication, SSH protocol is usually used. OpenSSH (OpenBSD Secure Shell) is a client/server application that is used to connect to Linux servers that implement SSH protocol. It provides encrypted communication between the client and the server, and was developed as part of the OpenBSD project.

Configuring and enabling OpenSSH on Linux on z Systems improves the communication security with the server when connecting to any system user ID. A simple setup requires just a user ID and the password to authenticate, but it is possible to implement two factor authentication, or use a public key infrastructure (PKI).

Some implementations of PKI allow users to connect to Linux servers without a password, requiring just the public and private key pair. A preferred practice is to set up PKI authentication using passwords. It prevents access to the Linux server in case an attacker has access to a user private key, but does not have the password.

Because encryption might be costly to system performance, throughput, or processor load, starting with OpenSSH version 4.4, it is possible to benefit from IBM z Systems hardware encryption to reduce the impact of expensive encryption operations.

For more information about OpenSSH configuration and enablement, see *Security for Linux on System z*, SG24-7728.

## **Network access to the insurance system**

When clients need to connect to the insurance system in this use case, the information, or data, that leaves the client's device and reaches the insurance system on a Linux on z Systems server needs to be protected against an attacker who wants to collect that data for visualization or to modify the data during the transmission.

Encrypting the communication between the client device and the application is usually used to avoid the data traffic in clear text. Most application servers use SSL/TLS to encrypt the communications with clients.

For IBM WebSphere Application Server security architecture, SSL/TLS is a key component. It is mainly used to protect HTTP traffic, IIOp traffic, LDAP traffic, MQ traffic, JDBC traffic, messaging traffic over the SIBus between WebSphere messaging engines, and J2C and SOAP traffic.

SSL/TLS is also supported in IBM DB2 databases using JDBC and SQLJ drivers. Using a secure DB2 database communication improves overall IT infrastructure security of the insurance company.

## **z/VM Connectivity**

Using z/VM network virtualization brings a layer of security to data being transferred to its guest systems. Queued direct input/output (QDIO) architecture uses direct memory access (DMA) to transfer network data from the address space of the sender to the address space of the receiver. When using HiperSockets networks, network traffic cannot be "sniffed" by other z/VM guests or other LPARs. The same happens with communication from OSA-Express to the guest, where network traffic is moved directly to/from the adapter to the address space of the sender/receiver.

With a local area network (LAN) configured on the hypervisor, you can establish communication through the guests without leaving the hypervisor layer. There are two ways to set up a LAN on z/VM. You can use Guest LAN, where a global LAN segment is defined and guests have a network interface card (NIC) connected to the guest LAN segment, or use z/VM Virtual Switch (VSWITCH).

A good practice is to disable TRANSIENT Guest LAN functionality as it allows user IDs defined with Privilege Class G to create their own dynamic Guest LANs, creating a possible breach of information flow control.

Use VSWITCH for LAN connectivity with the security options it offers. The transport of packets, when using VSWITCH, are done at the hypervisor layer, removing the processor burden of a guest needing to act as a router, bringing performance benefits as well.

VSWITCH supports Layer 3 (TCP/IP) or Layer 2 (Ethernet) modes and it is possible to define it to use the same subnet of the external network to which it is connected. It also supports IEEE 802.1Q VLAN, making the network management simpler with the use of VLAN subnets.

It is important to correctly set up the VSWITCH after the network infrastructure is planned. By default, VSWITCH does not allow guests to couple to it, so guests need to be explicitly allowed to couple to the VSWITCH. This helps control which guests have access to which VSWITCHes, but granting the access carefully also prevents guests from accessing LAN segments that they are not supposed to have access to. VLAN tagging helps keep control of this situation in addition to the creation of RACF/VM groups.

Using a VSWITCH access group and VLAN tagging can efficiently provide access control to system resources, but are considered discretionary rules that must be carefully maintained to avoid security breaches. The use of Mandatory Access Control (MAC) addresses the management problem. MAC is a security policy that governs which subject or users can access which resources, and in what way. MAC restricts access to an object by using the following main criteria:

- ▶ The security label assigned to the subject.
- ▶ The security label assigned to the object.
- ▶ The access the subject requires to perform a task.

If those three MAC criteria are met, z/VM proceeds with the discretionary access control when appropriated.

Using MAC to create security zones adds flexibility to system administrators when managing the system resources. It is implemented by activating security labels under RACF/VM. For more information about how to activate and set up security labels, see *Security for Linux on System z*, SG24-7728.

The use of VSWITCH for network management with RACF/VM for security and audit purposes gives z/VM hypervisor and guests a higher level of security while decreasing the time spent to manage it.

For more information about VSWITCH technology and its configuration, see the *z/VM Connectivity Guide*, SC24-6174.

## 3.2 Securing the enterprise applications

It is critical for the insurance company to only authorized requests that are processed by the CICS GENAPP application. To secure the connection from IBM MobileFirst to CICS in the example, SSL mutual authentication is used.

SSL/TLS is a well understood and popular way to provide encryption and data integrity. It can also be used to enable both server and client authentication.

Server authentication with CICS requires CICS to have an X.509 certificate stored in the CICS region's certificate key ring. The certificate (referred to as the CICS server certificate) is used as part of the SSL handshake processing. The client, in this case an IBM MobileFirst Server, validates the CICS server certificate. Successful server authentication requires that the certificate authority (CA) that signed the CICS server certificate be considered trusted by IBM MobileFirst. To be considered trusted, the certificate of the CA must be in the keystore of the client.



To use mutual authentication, the IBM MobileFirst Server must have a client X.509 certificate (called the IBM MobileFirst client certificate). The CICS region validates the IBM MobileFirst client certificate. Successful client authentication requires that the CA that signed the IBM MobileFirst client certificate be considered trusted by CICS. To be considered trusted, the certificate of the CA must be in the key ring of the CICS region.

CICS uses the z/OS system's SSL (a component of z/OS Communications Server) to support both the SSL and TLS protocols.

The following are additional considerations to secure your enterprise applications:

- ▶ Certificate and RACF user ID checklist
- ▶ Enable SSL
- ▶ Enable SSL mutual authentication
- ▶ Optimize SSL performance

For more information about these considerations, see Chapter 11 of *IBM System z in a Mobile World Providing Secure and Timely Mobile Access to the Mainframe*, SG24-8215.

### 3.3 Operating system security - z/VM

Because z/VM is a hypervisor and an operating system at the same time, most security practices that apply to a traditional operating system also apply to z/VM. However, some considerations need to be made when looking at it as a hypervisor.

The main consideration concerns what should be protected at the z/VM level. Planning must be done to protect all known points of failure. You would think that the virtual machines (guests) must be protected, but because z/VM is the hypervisor, protection of the Control Program (CP), the core of z/VM, needs to be in place. Data security is also a consideration and protection needs to be in place while it is transferred through the virtual network of z/VM, and when it is stored on a disk or on backup media.

Using z/VM as the hypervisor to virtualize an IT infrastructure enables the organization to manage the servers using universal management tools, reducing costs that are related to server management, and maximizing the utilization of IBM z Systems hardware.

Although z/VM can be installed alone as a unique logical partition (LPAR) on a machine to virtualize the z Systems hardware, you can also run z/VM in an LPAR and other operating systems on different LPARs, such as z/OS, IBM z/VSE® and Linux for z Systems. When used to virtualize the z Systems hardware, z/VM can either virtualize the same z Systems server it is running on, or emulate hardware features not necessarily available on its hardware.

It is possible to use z/VM to provide a customized environment for each guest, giving them z Systems features that are accessed simultaneously without the need of a real physical server for each guest. Alternatively, z/VM can handle access to the real devices for each guest as needed.

An advantage of using z/VM to virtualize the IT infrastructure is that the number of guests on z/VM is limited to the amount of resources available on the z Systems on which it is installed. In this case, the environment can be set up to use most hardware resources that z Systems can provide.

For more information about z/VM security, see *Security on the IBM Mainframe, Vol. 1 A Holistic Approach by Reducing Risk and Improving Security*, SG24-7803.

### 3.3.1 Virtual machine security

Because z/VM is a virtualization hypervisor, its main task is to handle multiple guests that share z Systems resources at the same time. It is important to keep guests' workloads separate, so a guest cannot threaten the integrity of either another guest or the hypervisor itself. This concept is called guest isolation and begins at the instruction level, so most instructions are run by z Systems hardware with little intervention from z/VM.

The Start Interpretive Execution (SIE) instruction on z Systems creates the environment for the guests. It handles the use of region, segment, and page tables that are set up by the z/VM Control Program for the guest. SIE instruction is used by z/VM to decrease the processor burden of the Control Program, and is available only with z Systems hardware.

Guest isolation only separates one guest from another. To secure the guests within a z/VM environment, define the scope of responsibility for each guest. A guest must only have access and authority to perform functions that are required by hosted workload. Therefore, always provide the minimum access and authority that are needed, no more, no less.

#### Privilege classes

The first step to limit access and authority for guests is plan the privilege classes. A guest can have one or more privilege classes defined to it. The privilege classes that the guest is assigned represent a job or role associated with that guest workload.

By default, z/VM has seven privilege classes, represented by letters from A through G. The default class with the least privilege is class G, known as the General User class. Classes greater than G are usually assigned to privileged users.

It is possible to create new privilege classes. This can increase the security of the z/VM virtual environment and must be in accordance with your guest role definition. These classes are created by the z/VM system administrator, and will meet the information security policy and the roles defined for guest workload. Custom privilege classes are represented using letters from I to Z, or the numbers 1 - 6.

There are cases that a command can affect more in the hypervisor than is expected for the guest. An example is the command ATTACH from Privilege Class B used to attach a real device to an ID in z/VM. A guest may need to issue the ATTACH command to be able to use a specific real device, but if this command is available in a custom class, the guest will be able to attach any real device defined in z/VM, meaning the guest can attach disks for other guests, as an example.

You can execute the COMMAND directory statement during logon processing. You can use this command to set up a user (similar to the ATTACH command) at logon time (LOGON, XAUTOLOG) but it does not allow you to "sporadically issue the ATTACH command" unless the user is also privilege class B. In the guest's user directory, you can specify the exact command the guest is authorized to issue. The following is an example COMMAND directory entry:

```
COMMAND ATTACH F000 TO LNXSRV10 AS D000
```

In this case, the guest is able to issue just the ATTACH command specified in its directory. It is able to only attach real device address **F000** to guest ID **LNXSRV10** as a virtual address of **D000**, thus avoiding the exposure of all other real devices defined in the z/VM hypervisor.

#### Logon-By

This LOGONBY directory entry z/VM feature is useful for system administrators. By using it in the directory for the guest definition, you can authenticate to this guest ID on z/VM using an

authorized z/VM user ID and its password. The following is an example of how the login command is used to authenticate to a guest:

```
LOGON LNXSRV10 BY WILLIANR
```

The password or password phrase for z/VM user ID WILLIANR is required to authenticate with the LNXSRV10 guest ID, and not the password or password phrase for LNXSRV10. This helps keep track of who accessed guests in case there is a team of administrators, providing accountability for access.

This functionality is extended to RACF/VM using the SURROGAT class of operation.

### 3.3.2 Hypervisor security

Using z/VM as a hypervisor gives Linux on z Systems valuable security and integrity features. Mainly, z/VM provides flexibility when consolidating servers while protecting the environment with multiple security zones. Running Linux as a guest on z/VM provides the best way to gain the maximum benefits from the z Systems platform.

The core functionality of z/VM is enforcing system integrity and guest isolation. It is important to review hypervisor configuration and customization before deployment. Sometimes the default configuration is not enough to comply with your organization's information security policy. A review of the z/VM system configuration file is a good start immediately after z/VM installation.

#### System Configuration file

The z/VM system configuration file, SYSTEM CONFIG, defines attributes of the z/VM system at IPL time. Within the z/VM system configuration file, you can customize basic default settings of the hypervisor to comply with an information security policy.

You can change the Privilege Classes commands that are described in "Privilege classes" on page 26. You might, for example, increase or decrease the privilege of a General User Class (Class G) or a System Operator Class (Class A) or even create a custom Guest Server Class (Class from I to Z, or 1 - 6) with specific commands allowed for use by the guest servers.

This is done using the MODIFY COMMAND and MODIFY DIAGNOSE statements in the z/VM system configuration file. These commands can redefine the existing CP commands and diagnose instructions from their default privilege class to another existing or custom privilege classes (among other things). The main advantage of these commands is to remove potentially dangerous commands from use before deployment.

As an example, the SHUTDOWN command is defined with the IBM default privilege class A, the System Operator class. This command can be moved to a new Privilege Class to avoid its use by mistake. The following command shows an example of how this can be redefined:

```
MODIFY COMMAND SHUTDOWN PRIVCLASS S
```

This command removes the SHUTDOWN command from privilege class A and places it in privilege class S. The SHUTDOWN command is used to shut down the entire z/VM LPAR. Because it is so disruptive, it is important that the information security policy restricts it to only the few roles that need to use privilege class S.

Another security setting that needs to be reviewed is enabling CLEAR\_TDISK on the FEATURES statement. With this feature enabled, the system will purge temporary disk space as soon as it is released back to the hypervisor. This will prevent guests from reading what was written in the temporary disk by other guests when the temporary disk is not used anymore and released.

There is a command defined in privilege class C that allows a user ID to escalate a privilege. This command is the SET PRIVCLASS. When a guest has access to privilege class C, this guest can use SET PRIVCLASS to escalate its privilege to any other privilege class defined at the z/VM system configuration file. This means that a guest with access to this command can issue any command at the hypervisor, escalating its privilege.

The z/VM system configuration file has a setting that allows the use of SET PRIVCLASS command, enabling or disabling it. The setting is called SET\_PRIVCLASS and must be reviewed before the z/VM deployment.

In the z/VM system configuration file, the use of the JOURNALING statement enables the system to log CP commands in the journaling records. This statement is useful when RACF/VM is not enabled on the system. Journaling options are covered in “Monitor system logs” on page 29.

## **The user directory**

The user directory holds the definitions for all virtual machines on z/VM. When a directory manager is not enabled, this file is updated manually and changes are put into place using the DIRECTXA CP command. To be able to issue this command, the user ID needs to have privilege class A, B, or C, so updates to the user directory are usually assigned to the system administrator role.

Without using an external security manager (ESM), user ID passwords and minidisks passwords are also manually managed in the user directory. In the machine-readable binary directory space, the passwords may be obfuscated, but they are not encrypted. Again, update of the user directory is usually assigned to the system administrator role so exposure is limited, but a system administrator is also the only person who can set passwords on the system.

Managing the user directory manually can be time consuming depending on the number of hypervisors and virtual machines. To reduce the time spent managing the user directory and avoid mistakes that can occur when it is done manually, enable a directory manager.

The IBM z/VM Directory Maintenance Facility (DirMaint™) is a collection of service virtual machines that control and manage changes to the virtual machine and hypervisor definitions. These virtual machines work as a focal point for directory-related operations that control the user directory.

Enabling DirMaint allows for an extra mechanism to protect virtual machine definitions. User directory statements can be added, deleted, or altered using DirMaint while it provides automated validation and extent allocation routines to reduce the chances of system administrator error.

Because DirMaint service systems have privileges through specific privilege classes to change the user directory, access to DirMaint commands must be controlled through its configuration files. Only system administrators or roles defined in the information security policy should have access to it.

## **Update management**

As described on 2.2.1, “Operating System Security - The Hypervisor” on page 13, keeping the operating system updated helps prevent security exposures. APARs labeled as SEC/INT are important to keep system security and integrity and should be applied as soon as possible.

Use of centralized patch management tools helps to keep track of updates applied and updates missing while controlling all systems that are managed by it.

Furthermore, an established update process keeps track of system updates and manages them in an acceptable time frame. This ensures a more secure environment for the guests.

## Monitor system logs

Customizing the z/VM system and keeping it updated are necessary steps to grant a more secure system, but are not the only steps related to security. Periodically monitoring the system, and keeping and examining audit trails must also be part of securing the hypervisor.

Most audit data comes from RACF/VM. It can audit every command and security-relevant event that happens within the hypervisor, in accordance with the information security policy.

When RACF/VM is not enabled, you can record actions on the hypervisor using the CP command journaling options. The journaling detects and records the occurrences of CP commands that it is configured to monitor. Analyzing the recorded information makes it possible to identify attempts to access a resource or log on the hypervisor.

The journaling options LOGON, AUTOLOG, XAUTOLOG, and LINK are responsible for recording information about the use of CP commands LOGON, AUTOLOG, XAUTOLOG, and LINK. It is possible to set up CP to take preventive actions based on the number of times that a user ID tries one of those CP commands. The following are the actions CP can take:

- ▶ Record the incident in the accounting data set
- ▶ Reject any new attempts to use that CP command (related to the journaling option)
- ▶ Lock the terminal for a certain amount of time
- ▶ Send a message to the user ID

## External security managers (ESMs)

An ESM is a key component of virtualization security at the hypervisor layer in modern environments. RACF Security Server (RACF/VM) is a priced ESM for z/VM that provides a centralized and effective security mechanism for authentication and authorization of users to access system resources.

Usually an implementation of a virtualized environment on z/VM without planning starts without the use of an ESM, but after a couple of Linux servers deployed it becomes clear that an ESM is required to support the environment. The following are some reasons to plan for using an ESM before deploying the guests:

- ▶ *Regulatory compliance:* Organizations that must comply with government and industry regulations for controlling and managing customer and client data require a level of security beyond that provided by just the z/VM internal security mechanism.
- ▶ *Audit capability:* An ESM is able to create clear audit reports that are required by government and industry customers. Manually creating an audit report can be time consuming without the use of an ESM.
- ▶ *Advanced security features:* Use of an ESM centralizes the security administration of all system resources. It provides access to detailed configuration attributes and a better control over the entire system.
- ▶ *Consistency across multiple z/VM systems:* It is possible to manage multiple hypervisors using a centralized ESM database to hold all security configurations. This configuration provides a high-level view of the entire virtualization infrastructure.

The following are advantages of using an ESM, among others:

- ▶ Overrides base CP security functions, giving more detailed options to control the security over system resources
- ▶ Encrypts virtual machine passwords and stores it on a database, replacing the user directory password
- ▶ Supports password phrases with mixed-case ranging from 14 to 100 characters in length
- ▶ Allows for greater control over CP commands, adding detailed options of control
- ▶ Allows for discretionary access controls (DAC) access list for resource use
- ▶ Allows for MAC security labels that group system resources

With RACF/VM it is possible to control privileged VM commands. It is possible to enable a resource class called **VMCMD** and create profiles to control the commands needed. The VMCMD class only controls a very small number of CP commands, and SHUTDOWN is not one of them. The list of commands controlled by the **VMCMD** class can be found at: [http://www-01.ibm.com/support/knowledgecenter/SSB27U\\_6.3.0/com.ibm.zvm.v630.icha7/authcp.htm?lang=en](http://www-01.ibm.com/support/knowledgecenter/SSB27U_6.3.0/com.ibm.zvm.v630.icha7/authcp.htm?lang=en).

Furthermore, RACF/VM allows for auditing of any security-relevant event on the system. Most auditing options on RACF/VM are disabled by default, so it is a good practice to list the control and auditing settings and enable all options that comply with the information security policy.

It is also possible on RACF/VM to create labeled security zones with *Mandatory Access Controls*. That way groups of system resources can be labeled and access granted to the necessary virtual machines. Without an access granted, a virtual machine cannot access a system resource defined at a security label even if this virtual machine has access to the *Access Control List* of that single system resource. In short, Mandatory Access Controls turn possible access to a resource into a denial of access to the resource.

This concept, referred as multi-tenancy, allows creation of multiple security zones within the hypervisor without the possibility of contamination between different virtual machines.

Enabling RACF/VM complements and extends basic z/VM security features while providing more granularity and accountability of all accesses and operations events. Its access control includes user verification, resource authorization, and logging capabilities. Enabling RACF/VM requires customization so that it can comply with your information security policy.

## **User ID and password management**

Whether or not you have an ESM enabled, it is a good practice to have user ID and password management rules in the information security policy.

Make sure the user IDs defined on the system comply with the user ID management tool used. It is a good practice to not allow the use of punctuation or symbols. Also, do not allow use of system keywords such as command names, command operands or 1 - 4 digit user IDs that might be spool IDs. Results using any of these options as user IDs are unpredictable.

For service machines that are not enabled or not used on the system, convert their directory password to NOLOG to deny the ability to log on to those machines. If RACF/VM is enabled on the system, revoke the IBMUSER user ID and remove all of its privileges.

Add to the information security policy the password rules for the user IDs. The following are some suggestions for password rules:

- ▶ Enforce changing the password when the user ID is first created with a default password.
- ▶ Enforce changing the password periodically (for example, every 180 days),
- ▶ Explore the use of passwords with mixed case and use of numbers.

A good password management policy helps prevent security incidents.

### **Improving security management**

Having RACF/VM enabled plays a vital role in helping to protect the hypervisor. When the number of hypervisor systems increase in a data center, even the use of RACF/VM as the centralized ESM can become complex and time consuming, requiring an increase of skilled staff.

IBM Security zSecure™ Manager for RACF z/VM can help system administrators on their daily tasks. It automates many recurring system administration functions like adding or deleting user IDs and groups, defining and granting access to users and groups, setting and resetting user IDs and passwords, and running daily or monthly reports. This reduces errors, minimizes complexity, improves quality of service, demonstrates compliance with the information security policy, and reduces costs.

Security zSecure Manager for RACF z/VM can be automated to repeatably audit the hypervisor and RACF/VM database, and create reports without the need of daily manual intervention from system administrator. This reduces the time that is spent by the system administrators on creating reports, and gives them time to analyze reports that automatically have security concerns highlighted and prioritized

The reports created by Security zSecure Manager for RACF z/VM also show missing or inconsistent definitions, decreasing the time to fix or prevent mistakes before they become a threat to security and compliance. It also includes a powerful system integrity analysis feature that can help reveal breaches in system integrity.

On the virtual machine side, Security zSecure Manager for RACF z/VM includes support for auditing events from Linux on z Systems. Auditing these event records helps detect Linux security threats and creates compliance reports for Linux systems.

## **3.4 Data security**

As discussed previously, for most of the organization, information is the most valuable asset, so protecting it is important for the organization's business. "Network access to the insurance system" on page 23 described how to improve the security of the data in flight, flowing through a network. It is also important to discuss the security of data at rest, the data that is stored on a disk or backup media.

Data protection starts at the planning stage and goes immediately through to daily use on production servers. The first place to define the accesses of a disk is during the definition, during the input/output configuration program (IOCP) set up. The IOCP definition handles the physical resources, including disks, to which an LPAR has access.

Depending on the number of disks, the LPARs on a CPC have access, the complexity of defining each disk to be visible to just a single LPAR increases and becomes costly. For this reason, the IOCP definition is usually done with all LPARs on a CPC being able to access almost every resource in the environment. The security is defined at a high level, with an ESM like RACF/VM, or an application security. Although this decreases the complexity during the

IOCP definition, it increases the complexity at a high level during RACF/VM definitions. It is a trade-off that needs to be decided during the IT infrastructure planning.

When using storage area network (SAN) volumes over Fibre Channel Protocol (FCP), techniques such as zoning and LUN masking are essential for controlling access to SAN volumes. This is especially important in a heterogeneous computing environment where devices from many manufacturers, using significantly different security architectures, all connect to the same fabric.

After the disk is defined and available to the LPAR, it is time to secure the minidisks. It is important to define minidisk access just to the users they are intended for. Information can be leaked between virtual machines when a minidisk has its access too lax, mainly if virtual machines from different security zones have access to it.

Although the minidisk password defined on the user directory can protect the access to the resource, it falls under the description of directory management in “The user directory” on page 28. The preferred practice is to use an ESM, like RACF/VM, to protect the resources. As already discussed, an ESM provides greater granularity to access authority, and extensive options for controlling auditing of both successful and unsuccessful accesses.

Controlling the access is also valid for shared disks. If you need to share files with multiple virtual servers, a preferred practice is to make this shared disk read-only and allow just one virtual machine to update it. This configuration reduces the risk of file corruption when multiple virtual machines try to write to it, and also the risk of a compromised virtual machine trying to place malware on the shared disk.

Protecting the access to the disks is important, but another way to improve data security is encrypting it. Because encryption depends on a key, proper handling and implementation of a key management policy is important. Failing on encryption key management can result in an encryption deadlock and permanent loss of all encrypted data.

The use of encryption on z Systems uses advantages by using cryptographic hardware and cryptographic functions that are built in the central processor, like the Central Processor Assist for Cryptographic Function (CPACF). It handles the cryptographic cipher calculations, leaving the central processor available for other uses and considerably reducing the central processor cycles when compared to the same cipher calculations done using software emulation.

Using tools to encrypt the Linux on z Systems can increase the data security. The dm-crypt subsystem in Linux is implemented as a device mapper that can be stacked on the top of other devices that are managed through the device mapper framework. You can therefore encrypt from entire disks to software RAID volumes and LVM logical volumes, adding flexibility to the encryption strategy.

For more information and steps on how to encrypt data on disks using dm-crypt, see *Security for Linux on System z*, SG24-7728.

Last but not least, remember that data on backup media must be encrypted. It is a security breach if a backup media with sensitive information leaves the data center and others outside the organization have access to that media. Protecting the data center and all devices within it is important, but allowing information to leave the data center without being protected is the same as ignoring all protection implemented in the organization data center.



## 3.5 Operating system security - Linux on IBM z Systems

Running Linux on IBM z Systems servers provides some advantage over a server farm physically distributed by using unique technologies used by the platform hardening the overall security. The following are security topics that need attention when implementing a Linux environment.

### Authentication

Authentication is the process of determining whether someone really is who he or she claims to be. The users attempting to access a system or a resource must first give sufficient proof of their identity. A server authentication is usually done using two of the following three categories, or factors for providing identity:

- ▶ Something that you know: A password or PIN
- ▶ Something that you have: A token, a user ID, a badge, a certificate
- ▶ Something that you are: Biometrics characteristics

The traditional way to authenticate on a Linux server is having a user ID on the system and knowing a password for it. Implementation of more than two of the factors is already available for Linux system, making it possible to request a user ID, a password, and a PIN that is generated by some electronic device, like a token or an application on a cell phone, when authenticating. Use of more factors for authentication brings a high level of security to what is being accessed.

The Pluggable Authentication Modules (PAMs) can be used to reinforce compliance with the organization information security policy by increasing the number of factors used to authenticate and only allowing access to user meeting the specific characteristics defined with PAM. Applications that are enabled to use PAM can be plugged into new technologies without modifying the existing applications. This flexibility provides administrators with these advantages:

- ▶ Use any available authentication service for an application
- ▶ Use multiple authentication mechanisms for a service
- ▶ Add authentication service modules without having to modify the application
- ▶ Use a single password for authentication on multiple modules

Another factor that improves security level is the use of PKI, such as SSH key pair. The users need to have a public and private key pair that ensures the user ID. Although it is possible to use SSH key pair without a password set to it, a password should be set to the key pair. This prevents an attacker who has access to the private key but does not know the password for the user ID from being authenticated at the server.

### Access control

Defining each job role in Linux is complicated because everything converges to the root user ID. However, doing so is a good practice that defines the access and provides strong control over who can access a superuser account.

The DAC model, which is the standard Linux security, does not provide protection from broken software or malware running as a normal user or root. Users can grant risky levels of access to files they own.

Use of MAC provides full control over all interactions of the software. Administratively defined policy closely controls users and process interactions within the system, and can protect the system from broken software or malware running as any user.

Security-Enhanced Linux (SELinux) on Red Hat Enterprise Linux (RHEL) is an implementation of MAC using Linux Security Models that is based on the principle of least privilege. When enabled in permissive mode, every access to a system resource by a user or process such as an I/O device must be controlled by SELinux. This can sometimes cause extra processing cycles on the system.

AppArmor is a Linux application security framework included with Novell SUSE Linux Enterprise and is an open source project. It takes a different approach from SELinux that provides an easy-to-use way for security applications at Linux. Following are some features that can be found at AppArmor:

- ▶ Yet another Setup Tool (YaST), which is an administration tool for configuration, maintenance, and automated development of a per-program security policy
- ▶ Predefined security policies for standard Linux programs and services
- ▶ Robust reporting and alerting capabilities to facilitate regulatory compliance
- ▶ Common Interface Model (CIM), which is a schema for clients that integrate with industry standard management consoles
- ▶ ZENworks Linux Management integration for profile distribution and report aggregation
- ▶ Path-name-based system that does not require labeling or relabeling of file systems

For more information about how to set up SELinux or AppArmor at Linux on z Systems, see *Security for Linux on System z, SG24-7728*.

## User management

When the IT infrastructure increases in its size, the number of user IDs and the complexity of managing the user IDs increase. It is important to have a way to manage user IDs, mainly for a security concern.

Centralizing the repository of user IDs helps in the management activities, reducing the administration effort compared to distributed user IDs management. It is considered a good practice for the maintenance of the information security policies that are applied to user management.

The centralization of user ID management involves adding, deleting, changing account information, and resetting passwords. Doing that from a single and centralized point, such as an LDAP server, can help keep the security requirements and policies consistent throughout the IT infrastructure. This configuration avoids the need to spread sensitive information from users, like passwords, to all servers.

When using a centralized user ID management server, all servers must connect to it using an encrypted connection. Not all of the information flowing between the LDAP server and the servers on the IT infrastructure is sensitive, but enabling this protection is simpler to implement than using a mix of encrypted and non-encrypted connections.

## Update management

Keeping the operating system updated helps prevent its exposure. An established update process under the servers keeps track of system updates and manages them in an acceptable time frame. Using a minimal system installation also helps keep control of security and system update. There are fewer potential points of security exposure when fewer packages are installed and managed.

The use of a centralized patch management tool can decrease the complexity and time spent to apply server patches when the number of servers being managed increases. It also helps

to keep track of patches applied and patches needed to all servers managed, avoiding the possibility to leave a server without the updates.

## **Audit**

A well-defined information security policy is worthless if there is no way to assess whether the policies are effective, meaning it was adhered by all employees and they are playing the roles that they are expected to.

Keeping track of changes, and authorized and unauthorized accesses is a way to make sure the information security policy is followed. But again, with the increase of servers managed on the IT infrastructure, the amount of audit data generated makes it impossible for a human to analyzed all of it, find a threat, and act on it while the intrusion is still happening. For that reason, define, during the planning stage of the IT infrastructure, which actions must be logged for audits.

The complexity in auditing is reduced when well-defined roles are available in the information security policy. Users under one role should not have access to override the mandatory access controls and should not be able to manipulate the controls that are under the jurisdiction of another job role. With the separation of duties, the functionality of the systems and integrity of audit logs will not be compromised.

To create a separation of duties under Linux, use SELinux or AppArmor. If those tools are not used or enabled, the task to control and determine user privileges become more complicated. A preferred practice then is to use Sudo to control access to privileged commands. Use of Sudo ensures better protection by limiting the privileged commands that a user can issue and protecting the root password from being shared with system administrators. Use of Sudo also ensures audit of accountability for users who issue privileged commands.

## **Cryptographic hardware**

Linux on IBM z Systems can benefit from the use of z Systems cryptographic hardware. It supports the use of CPACF and Crypto Express5S (the latest available at the time of writing) by using in-kernel crypto APIs and the libica cryptographic functions library. Use of these features provide these benefits:

- ▶ File system encryption
- ▶ Communication encryption (to the applications like IBM HTTP Server)
- ▶ System security by providing advanced cryptographic functionality

CPACF is available on every processor unit defined as a central processor (CP) and can be explicitly enabled by using the enablement feature #3863, for no extra fee. It provides a set of symmetric cryptographic functions that enhance the encryption and decryption performance of clear-key operations for SSL, virtual private network (VPN), and data storing applications not requiring a high level of security, such as Federal Information Processing Standards (FIPS) 140-2 Security Level 4, when compared to the same operations done using software emulation encryption.

The CPACF coprocessor on the IBM z13 has been redesigned and has better performance when compared to the zEC12.

CPACF offers the following data encryption and decryption algorithms for data privacy and confidentiality:

- ▶ Data Encryption Standard (DES):
  - Single-length key DES
  - Double-length key DES
  - Triple-length key DES (TDES)
- ▶ Advanced Encryption Standard (AES) for 128-bit, 192-bit, and 256-bit keys

CPACF offers the following hashing algorithms for data integrity:

- ▶ SHA-1: 160 bit
- ▶ SHA-2: 224, 256, 384, and 512 bit

For MAC, CPACF offers these options:

- ▶ Single-length key MAC
- ▶ Double-length key MAC

For cryptographic key generation, CPACF offers Pseudorandom Number Generation (PRNG) algorithms.

Crypto Express5S is an optional feature that was not available in previous generations and is designed to complement the cryptographic capabilities of the CPACF. It is in the Peripheral Component Interconnect Express Generation 2 (PCIe Gen2) I/O drawer and can be configured in one of the following three ways:

- ▶ Secure IBM Common Cryptographic Architecture (CCA) coprocessor (CEX5C), supporting:
  - Secure key functions
  - FIPS 140-2 Security Level 4 certification
  - User Defined Extension (UDX) services to implement custom cryptographic functions and algorithms.
- ▶ Secure IBM Enterprise Public Key Cryptography Standards (PKCS) #11 (EP11) coprocessor (CEX5P), providing:
  - Open, industry-standard cryptographic services following the PKCS #11 specification v2.20 and more recent amendments. It was designed to extended FIPS and Common Criteria evaluations to meet public sector requirements. The new cryptographic coprocessor mode introduced the PKCS #11 secure key function.
- ▶ Accelerator (CEX5A), optimized for:
  - Public and private key cryptographic operations, used with SSL/TLS processing.

For more information about CPACF and Crypto Express4S, see the *IBM zEnterprise EC12 Technical Guide*, SG24-8049. For more information about how to use those features at Linux on z Systems, see *Security for Linux on System z*, SG24-7728.

## Firewall

The use of a firewall is defined by the IT Infrastructure and by the information security policy of an organization. Using the guest isolation feature under z/VM and z Systems architecture makes such a solution as secure as having a firewall running on every Linux on z Systems. However, if the information security policy enforces the use of a firewall on any backend server, including those running on z Systems environment, that should be implemented.

There are several sophisticated firewall features solutions available for Linux that can filter and manipulate packets based on complex rules defined by the system administrator. A preferred practice is to use restrictive firewall policy instead of permissive policy. This ensures that packets that are explicitly not allowed to flow to the network are dropped, instead of rejected.

Some tools help to automate firewall policy creation. SUSE Enterprise Linux offers a firewall configuration tool using YaST that can be used both in graphical mode or text mode. Red Hat Enterprise Linux offers a firewall configuration tool called system-config-firewall that can also be used in graphical or text mode. Another option is a tool called Firewall Builder. It is an open source tool and can be found at:

<http://www.fwbuilder.org>

Firewall Builder can be downloaded and built for SUSE Enterprise Linux Server or Red Hat Enterprise Linux.







# End to End Security with z Systems



## Understand z Systems end-to-end security

## Secure z Systems for z/OS, z/VM and Linux for z Systems

## Understand security from a use case perspective

This IBM Redpaper Redbooks provides a broad understanding of the components necessary to secure your IBM z Systems environment. It provides an end-to-end architectural reference document for a use case that employs both mobile and analytics. It also provides an end to end explanation of security on z Systems from the systems of record through the systems of engagement. Security is described in terms of transactions, covering what happens after a transaction hits the system of engagement and what needs to be in place from that moment forward.

The audience for this paper is IT architects and those planning to use z Systems for their mobile and analytics environments.

## INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

## BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:  
[ibm.com/redbooks](http://ibm.com/redbooks)