Security with
Linux on z Systems and LinuxONE
- Selected Aspects

Dr. Manfred Gnirss

IBM

Please note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

© Copyright IBM Corporation 2017. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Notice and disclaimers

Copyright © 2017 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.**

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

© Copyright IBM Corporation 2017. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

Notice and disclaimers cont.

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 in connection with this publication 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. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products**. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular, purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, AIX, BigInsights, Bluemix, CICS, Easy Tier, FlashCopy, FlashSystem, GDPS, GPFS, Guardium, HyperSwap, IBM Cloud Managed Services, IBM Elastic Storage, IBM FlashCore, IBM FlashSystem, IBM MobileFirst, IBM Power Systems, IBM PureSystems, IBM Spectrum, IBM Spectrum Accelerate, IBM Spectrum Archive, IBM Spectrum Control, IBM Spectrum Protect, IBM Spectrum Scale, IBM Spectrum Storage, IBM Spectrum Virtualize, IBM Watson, IBM Z, IBM z Systems, IBM z13, IMS, InfoSphere, Linear Tape File System, OMEGAMON, OpenPower, Parallel Sysplex, Power, POWER, POWER4, POWER7, POWER8, Power Series, Power Systems, Power Systems Software, PowerHA, PowerLinux, PowerVM, PureApplica- tion, RACF, Real-time Compression, Redbooks, RMF, SPSS, Storwize, Symphony, SystemMirror, System Storage, Tivoli, WebSphere, XIV, z Systems, z/OS, z/VM, z/VSE, zEnterprise and zSecure are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Linux is a registered 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.

© Copyright IBM Corporation 2017. Technical University/Symposia materials may not be reproduced in whole or in part without the prior written permission of IBM.

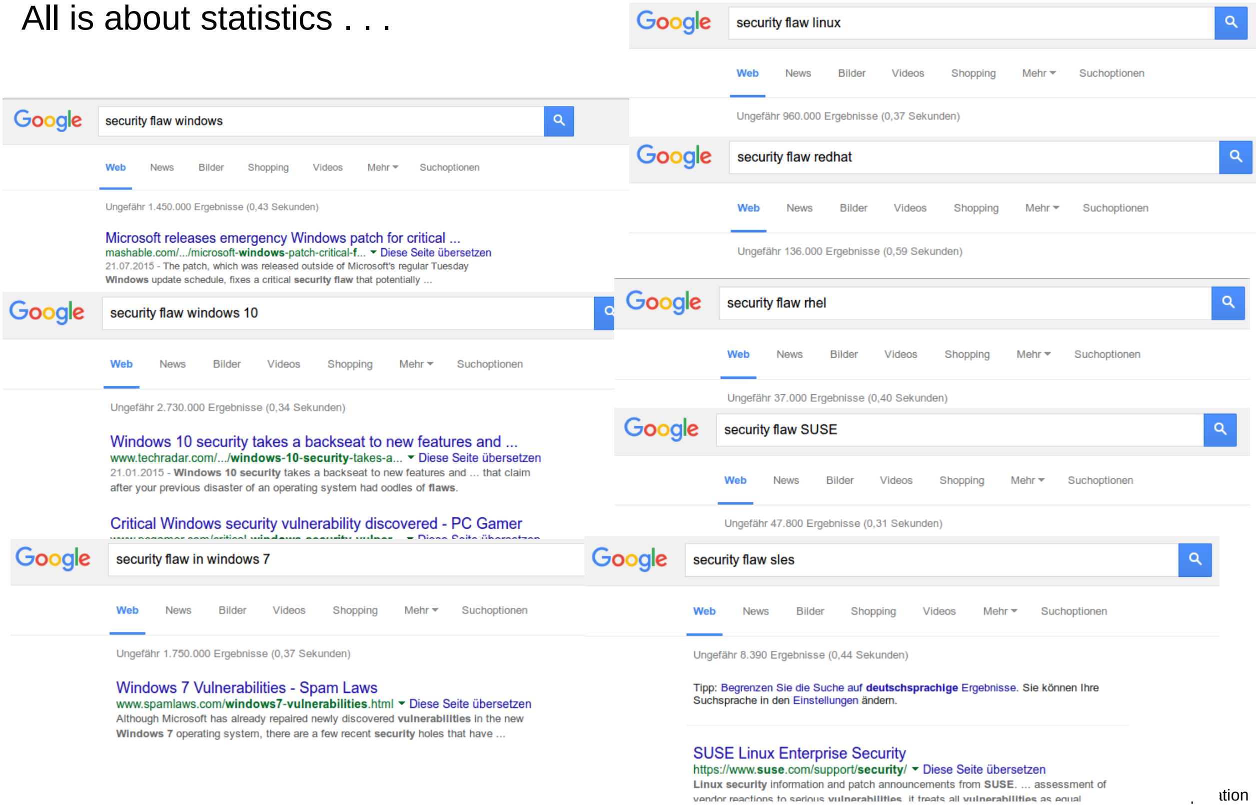
Agenda

- Philosophy: open source vs. closed source
- z Systems – Hardware
- Virtualization – z/VM
- Linux
- Linux for z Systems (incl. KVM and Docker)
- Standards and Certifications
- Linux Auditing
- Cryptography with hardware support of z Systems
- Container – Docker
- Centralized Audit with ITDS and RACF

© *2017* IBM Corporation

Philosophy: Open Source vs. Closed Source code

All is about statistics . . .



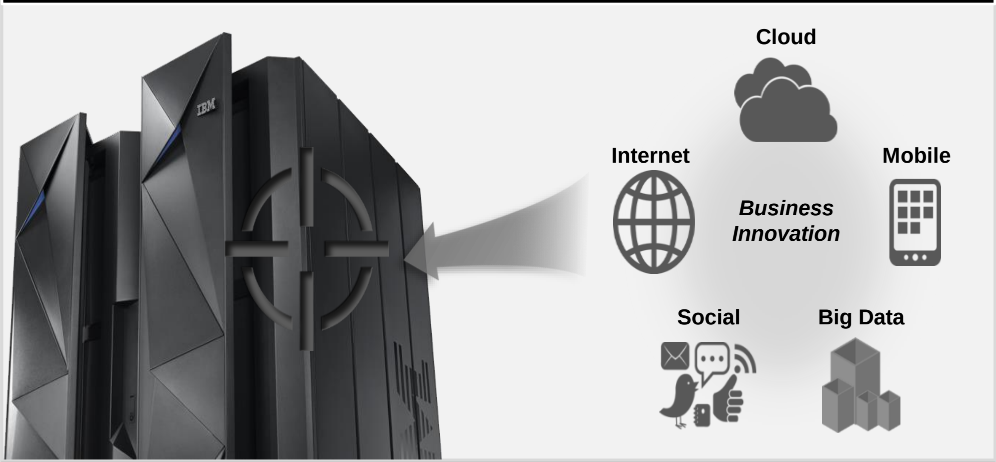
ıtion

The increasingly desirable target of the mainframe

80% of all active code runs on the mainframe

80% of enterprise data is housed on the mainframe

Today's technologies are eliminating "mainframe isolation"

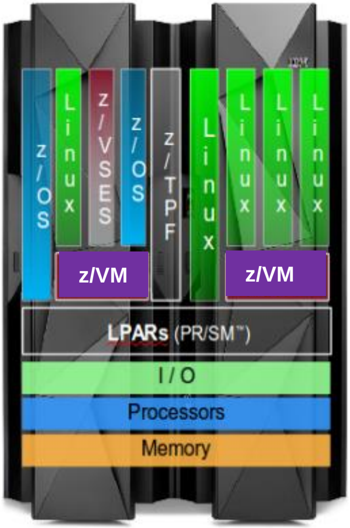


© 2017 IBM Corporation

IBM LPAR technology and IBM z/VM

LPAR (PR/SM)

- Logical partitions (hardware level)
- "independent" environments
  - Production
  - Test
  - Education
  - Demilitarized zone
  - …
- Common Criteria certified by BSI

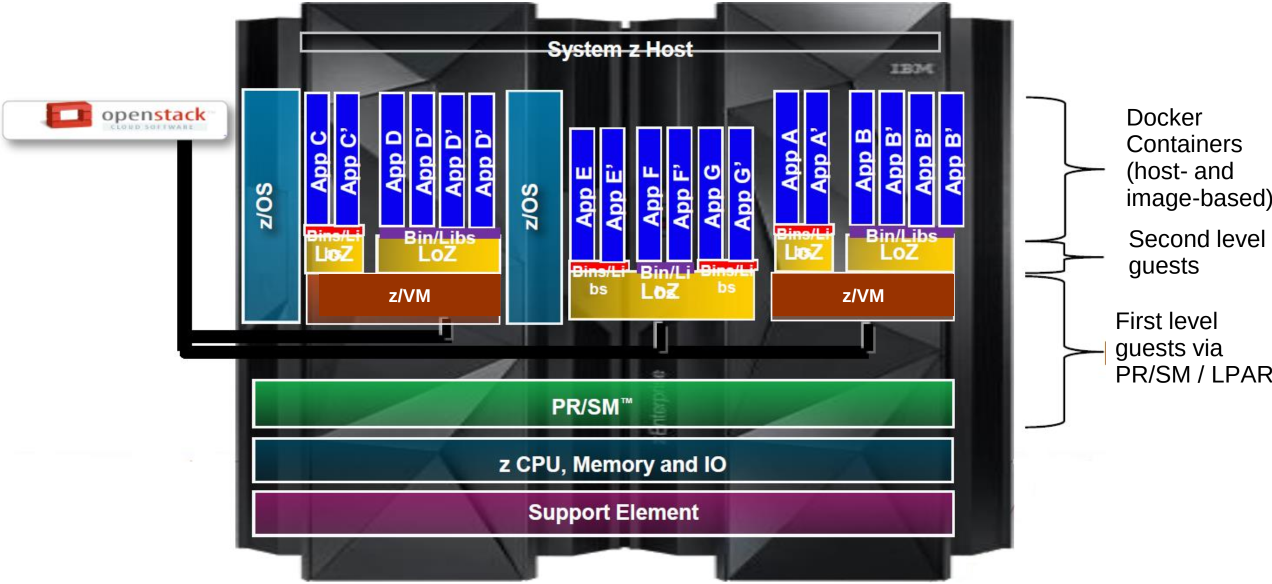


z/VM is IBM's virtualization

- World class quality, security, reliability - powerful and versatile
- Extreme scalability creates cost savings opportunities
- Exploitation of advanced technologies, such as:
  - Shared memory (Linux kernel, executables, communications)
- Highly granular control over resource pool
- Valuable tool for resiliency and Disaster Recovery
- Provides virtualization for all IBM Z operating systems

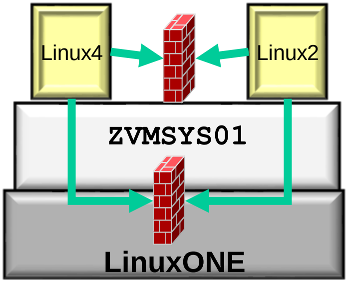
© *2017* IBM Corporation

Sample on Shaping
a Docker-based Environment on IBM Z



Security and Integrity on each level is needed:
Different requirements?

Guest Isolation on z Systems and LinuxONE

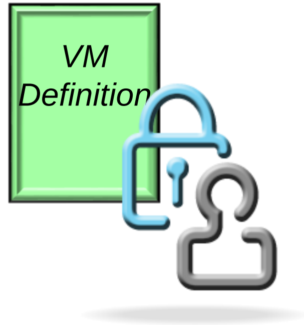


- All guests must be isolated from one another
  - Separation of duties and need to know
  - Control the flow of data
  - Keep workloads from interfering with one another
- Isolation on IBM Z & LinuxONE starts at hardware
- The **Interpretive Execution Facility** and **Start Interpretive Execution (SIE)** instruction are how virtual machines are executed
  - PR/SM controls LPAR creation
  - z/VM Control Program (CP) controls VM instantiation
- SIE instruction "runs" a virtual machine until a condition is raised
  - "What happens in a VM stays in a VM"
  - No mechanism for hyperjacking the platforms
  - Only leaves machine on interception conditions (a.k.a. "SIE break")

© 2017 IBM Corporation

Scope of Responsibility

- Any virtual machine is constrained in its ability to impact the hypervisor
  - Role-based access controls
  - Administrator vs. general-use commands
  - Communication with other machines / resources

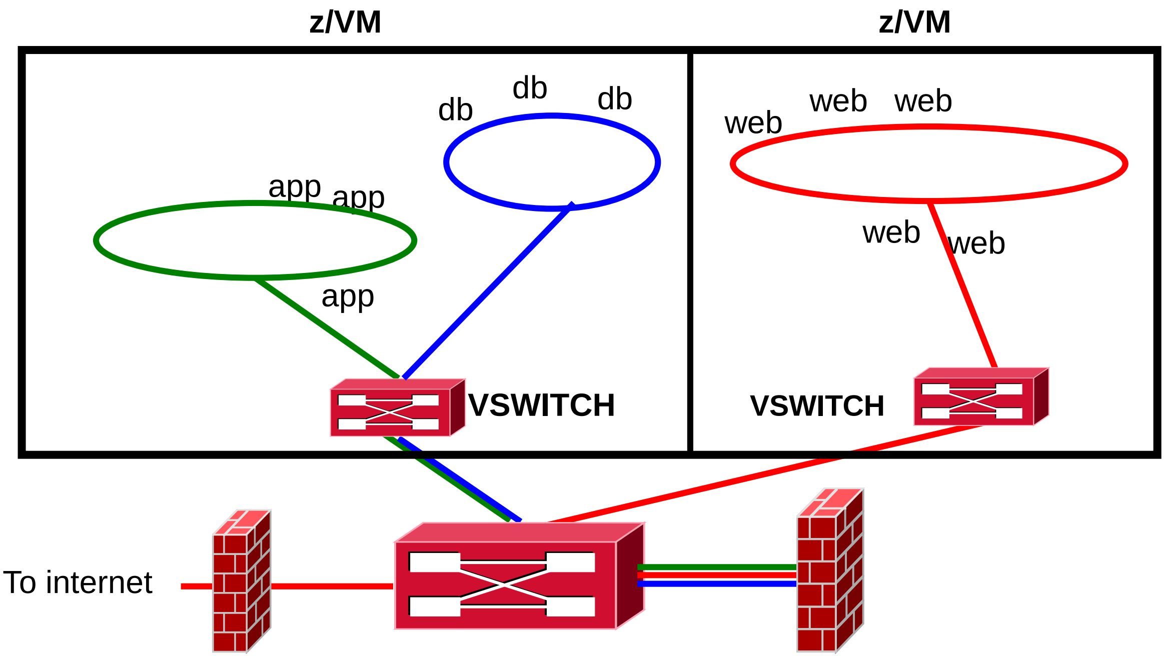


z/VM

- **Privilege classes** (Class G or less)
  - Administrators can write their own classes
  - SVMs and Operators may have more
- **Directory statements** to augment VM definitions:
  - LOGONBY statement for controlled access
  - COMMAND statements for pre-LOGON context creation
  - CRYPTO statement for z Systems CryptoExpress access
  - LINK and NICDEF for controlled access to virtual resources

© 2017 IBM Corporation

Virtual Switches, VLANs, and Zoning



Hypervisors, by their natures, are highly flexible

- There are a lot of options to consider
  - Alternate communication paths to check
  - Virtual networking options to control
  - Shared memory spaces
  - Access to data at rest (storage, tape)

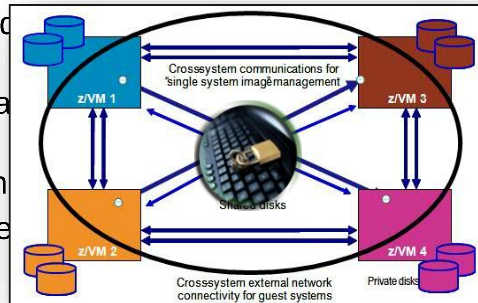
- And other considerations to factor in …
  - Password controls – are they in the clear? Are they changed?
  - Auditing – are you logging the right security-relevant events? Can you?

A security manager provides both a finer granularity of control and the ability to enact more complete isolation of guests and projects … in a consolidated interface.

© *2017* IBM Corporation

Infrastructure Security with RACF for z/VM

- RACF Security Server is a priced feature of z/VM
- A **requirement** for meeting today's enterprise security requirements
- RACF enhances z/VM by providing:
  - Extensive auditing of system events
  - Strong Encryption of passwords and password phrases
  - Control of privileged system comma
  - Extensibility in z/VM environments clustered through Single System Im
  - Controls on password policies, acce rights, and security management
  - Security Labeling and Zoning for multi-tenancy within a single LPAR (or across a cluster)

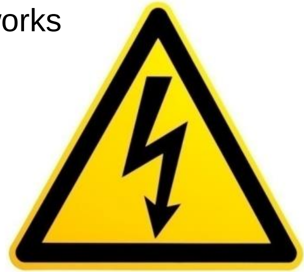


RACF for z/VM is an **integral component** of z/VM's
Common Criteria evaluations (OSPP-LS at EAL 4+)

Example* risks to sensitive data in virtual environments

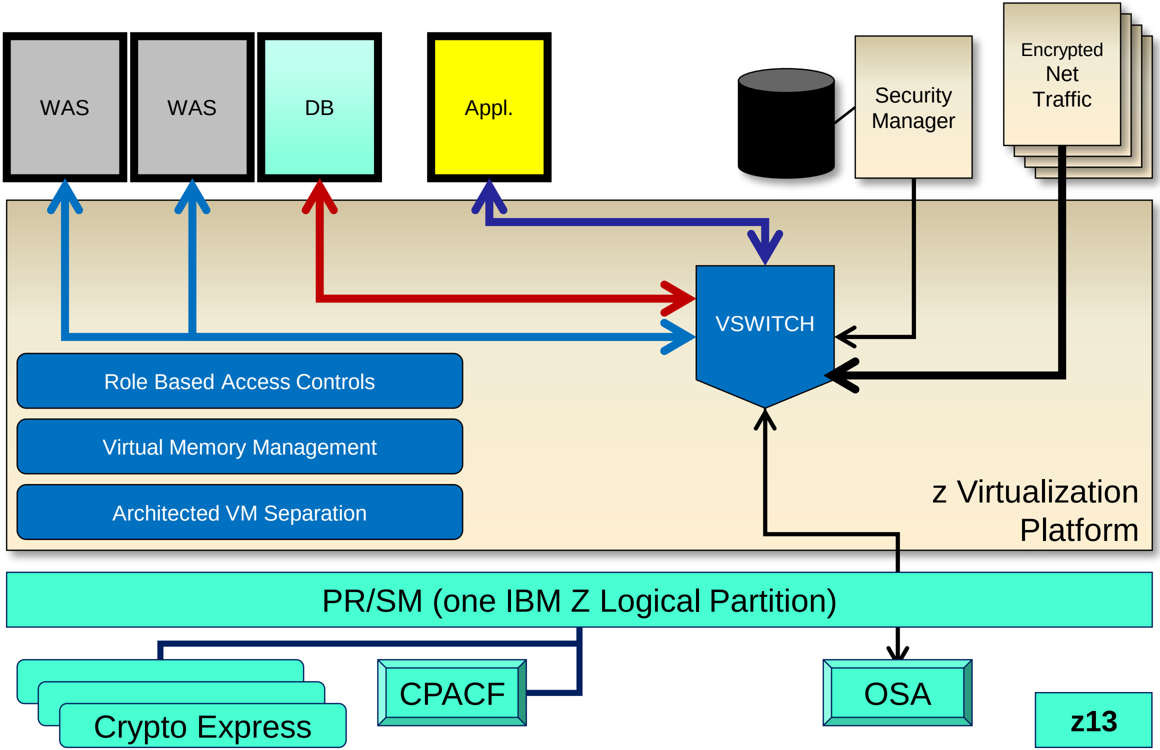
***(PCI DSS v3.1 Supplement - Virtualization Guidance v2.1)**

1. Vulnerabilities in the Physical Environment Apply in a Virtual Environment
2. Hypervisor Creates a New Attack Surface
3. Increased Complexity of Virtualized Systems and Networks
4. More than One Function per Physical System
5. Mixing VMs of Different Trust Levels
6. Lack of Separation of Duties
7. Dormant Virtual Machines
8. VM Images and Snapshots
9. Immaturity of Monitoring Solutions
10. Information Leakage between Virtual Network Segments
11. Information Leakage between Virtual Components



© *2017* IBM Corporation

This is your LinuxONE System On Lockdown



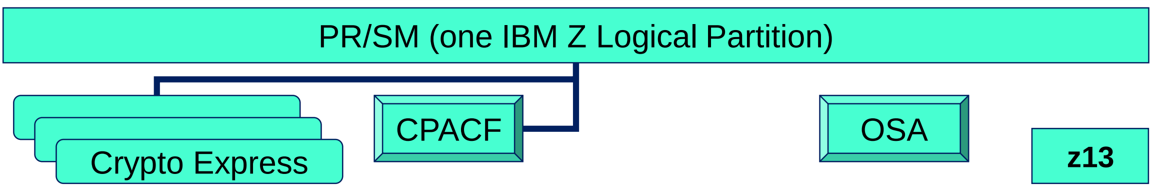
© 2017 IBM Corporation

Linux Guest Security

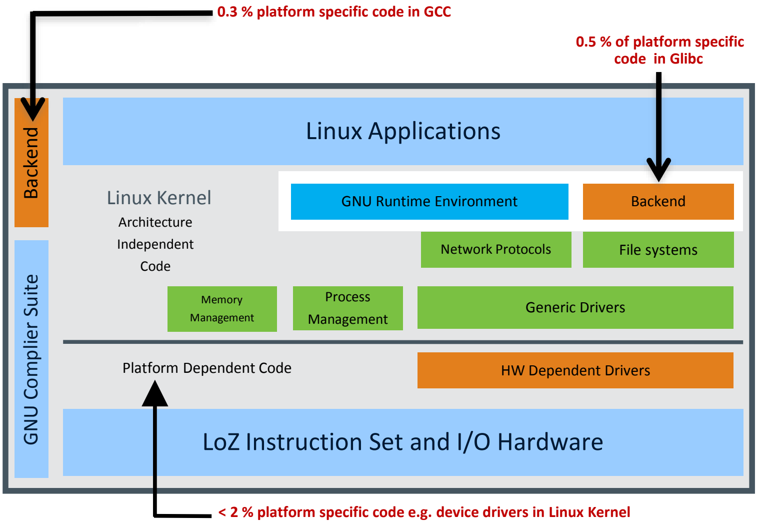
Linux04

Of course, all of the preceding content assumes
you will secure your Linux guests
with the same diligence and vigilance
as you do your hypervisor.

It does no good to lock the door
if you leave the window open.



Linux is Linux is Linux – even on the Mainframe!



- Port of the open source GNU / Linux® operating system to IBM z Systems™ architecture (including optimization)
- Pure Linux – it's an ASCII environment like all other Linux too
- Conforms to the Linux Standard Base (LSB)
  - Based on and compliant to the POSIX specification and several further open standards

- **NO emulation** - natively exploits z Systems hardware
  - Runs native in a LPAR or virtualized under z/VM®

Design principles of Linux for z Systems:

- **Not a unique version of Linux**
  - No changes to the standard kernel
- **No changes regarding Look & Feel**
- **Not a replacement for another IBM operating system**

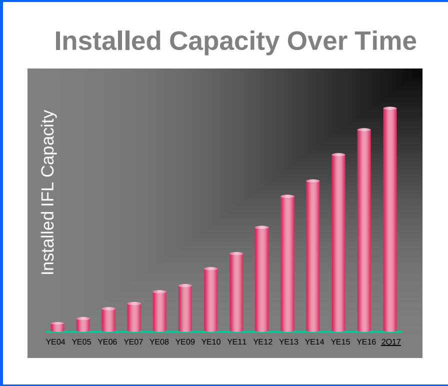
Many Linux software packages did not require any code changes to run Linux on z Systems!

Linux on IBM z Systems in 2Q2017

Installed Linux MIPS at 40% CAGR*

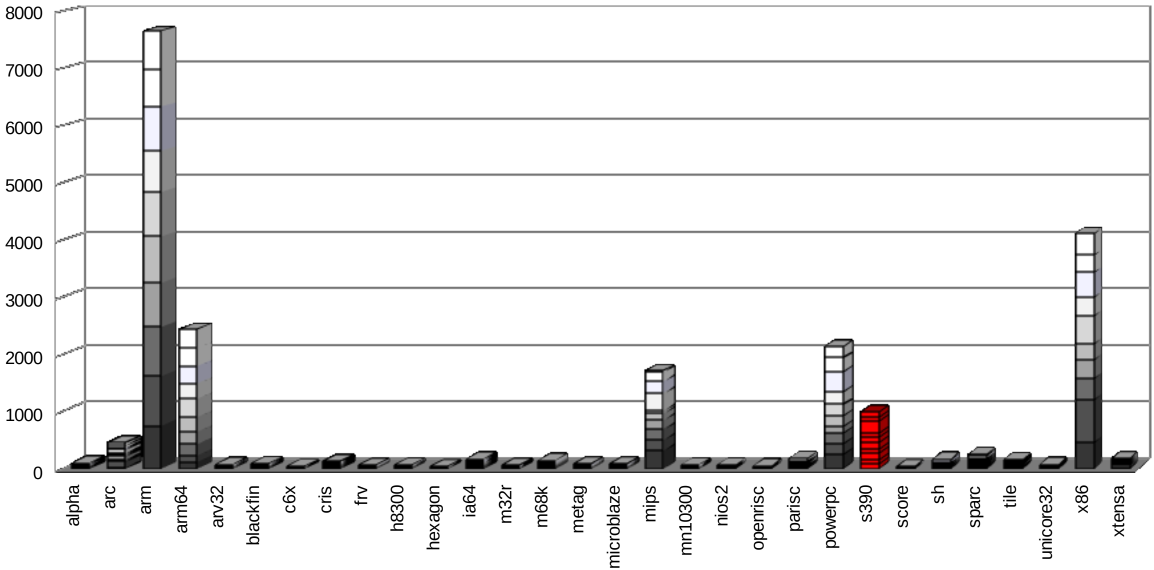
- 29.9% of Total installed MIPS run Linux as of 2Q17
- Installed IFL MIPS increased by 21% YTY from 2Q16 to 2Q17
- 49% of IBM Z Enterprises have IFL's installed as of 2Q17
- 90 of the top 100 IBM Z Enterprises are running Linux on z as of 2Q17 **
- 37% of all IBM Z servers have IFLs
- 60% of new FIE/FIC IBM Z Accounts run Linux

** Based on YE 2003 to YE 2016   **Top 100 is based on total installed MIPS*



© 2017 IBM Corporation

Git commits per architecture in 4.x



© 2017 IBM Corporation

Linux on z is Linux ... With all of z's Benefits

- Linux is Linux
  - Linux security features and tools available to all architectures
  - Differences only in
    - architecture specifics
    - device support

- Thorough open source review of key components
  - Security is and was always a focus of kernel development
  - Core Infrastructure Initiative (a.o. sponsored by IBM) focuses on supporting security relevant packages (like openSSL)
  - OpenMainframe project: community involvement

- Benefits stem from the platform
  - Strong guest isolation
  - Cryptographic hardware support

- Linux for z Systems advantage
  - Probably not a good target for low focus attacks

© 2017 IBM Corporation

Linux is Linux . . .

- Linux is Linux
  - Linux security features and tools available to all architectures
  - Differences only in
    - architecture specifics (e.g memory management / CPU layout)
    - device support (crypto)
- Thorough open source review of key components
  - Security is and was always a focus of kernel development
  - Heart-bleed Core Infrastructure Initiative (a.o. sponsored by IBM) focuses on supporting security relevant packages (like openSSL)
- The Linux for z Systems advantage:
  - small number of users
    - **not a good target for low focus attacks**
  - requires special skills & resources
    - **probability of low skill attack of Linux on z is low**

© 2017 IBM Corporation

Linux is Linux . . .

… but features, properties and quality
depends on the underlying architecture

- QoS, Redundancy or RAS features build-in hardware
(Redundant Array of Independent Memory / RAIM, outage avoidance using hotplug hardware)

- Hardware supported large scale virtualization support
(highly efficient, granular and isolated virtualization that is part of the architecture by design)

- System features
(Business Continuity using GDPS / xDR, I/O bandwidth, Capacity on Demand / CoD, autonomic Workload Management / WLM, Hipersockets, Power Capping)

- System's workload characteristics
(small/discrete, highly threaded, parallel data structures, shared data and work queues, mixed workload)

- Hardware requirements / availability
(Crypto: CPACF / Crypto Express, Decimal Floating Point / DFP,
3D GPU cards, USB dongles, ASIC cards)

- Operating system or software requirements / availability
(for example the IBM Communication Controller is available for Linux on z Systems, but not for Linux on Intel ↔ DB2 LUW Express-C is available for Linux on Intel, but not for Linux on z Systems)

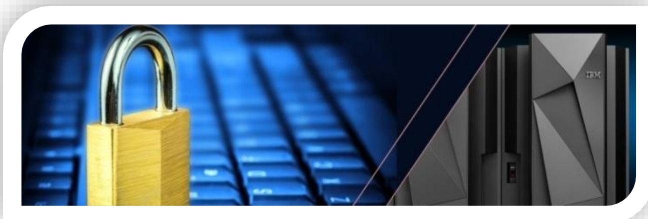
- Licensing constraints
(Sub-capacity options, usually close related to virtualization support)

© 2017 IBM Corporation

Linux on z and Open Source Security

(the starter list)

- **SELinux** for access control (see also: **AppArmor**)
  - A foundational component of Linux security
  - Used to define policies for security within a Linux guest
- **sudo** and **cgroups** for resource control — **<< you didn't give out root, right?**
- **openLDAP** for open identity management — **<< or find a SAML solution**
- **openSSL** and **openSSH** for secure communication — **<< don't forget httpd.conf**
- **IPtables / NetFilter** for firewalls — **<< if you run them inside z …**
- **dm-crypt / LUKS, eCryptFS** for file-system encryp — **<< we'll cover this soon …**
- **Lynis, Tiger,** or **openSCAP** for system hardening — **<< measure your guest vs baselines**



Linux - Malware




- **Antivirus**
  - Today, the thread of viruses for Linux systems can be considered as relatively low. But Linux for z Systems should not distribute (Windows-) viruses.
  - On Linux for z Systems: ClamAV
    - For SUSE: ClamAV is part of the distribution (incl. Service), Download the signatures from the net (may be with a cron job)
    - For Redhat: Download ClamAV from the ClamAV site
- **Rootkits**
  - Serious thread, hard to detect and difficult to remove
    - Replacement of user application with modified program.
    - Installed as Linux Kernel Module (LKM) – can modify syscalls – you can't trust the kernel anymore!
  - Prevention and Monitoring:
    - Monitoring by integrity checking (look for changes, fingerprint) – Example: Tripwire
    - Prevent from install, no root access to attacker....
    - the best way to prevent rootkits is by practicing smart security, for example, firewalls, good passwords, checking permissions etc. - Practicing good security, for example, using SELinux
    - The rootkits which are unknown and use LKM are one of the worst ones a Linux user can get.
  - Removal: New system installation (backup might be already compromised)

© *2017* IBM Corporation

Security policies

- Ensure that there is a Security Policy in place . . .
  - Authentication policies (e.g. password rules)
  - Network access connectivity (firewalls)
  - Authorization policies (roles)
  - Auditing policies
  - How to use root account
  - Security Compliance requirements
  - Apply security patches and updates
  - Both Red Hat and SUSE are committed to fix security exposures in a timely manner
  - You must apply security maintenance ASAP!!!
  - Every system connected to the outside world that has not the latest security fixes applied risks being broken into.….

  - For example check for updates:
    - z/VM PTFs

    http://www.vm.ibm.com/security/

    - Linux on z Systems prereqs: http://www.ibm.com/developerworks/linux/linux390/distribution_hints_z13.html

© 2017 IBM Corporation

Security Standards

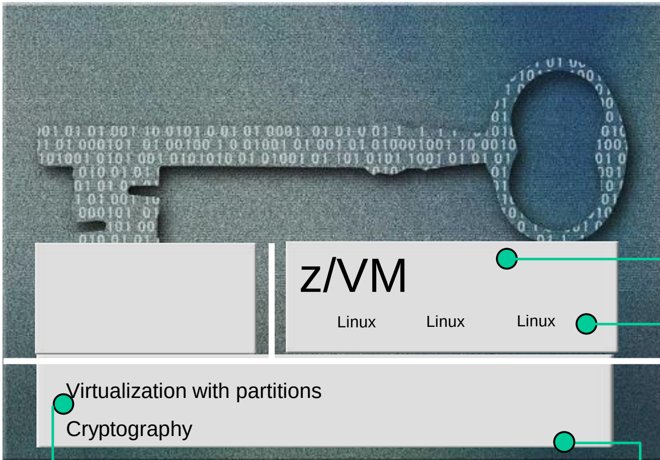
- Increasing importance of regulations and compliance of security standards.
- Some standards:
  - Comon Criteria with Operating System Protection Profile (OSPP)
  - Payment Card Industry Data Security Standard (PCI-DSS)
  - GDPR (General Data Protection Regulation)
  - BSI (Grundschutzkatalog)
  - Bundesdatenschutzgesetz, Sozilagesetzbuch, IT Sicherheitsgesetz
  - HIPAA
  - SOX
  - BASEL II
  - Solvency
  - . . .

- Idea: Even for the case that not mandatory for your IT environments, standards are good orientation to think about security . . .

© 2017 IBM Corporation

Security Certifications

The IBM z14 . . .
Crypto Express6S . . .



z/VM

- Common Criteria
  - z/VM 6.3 is certified EAL4+ level for OSPP and FIPS at the 140-2 level
- Statement of Integrity

Linux on z Systems

- Common Criteria
  - SUSE SLES11 SP2 and SLES 12 certified at EAL4+ with OSPP
  - Red Hat EL6.2 / 7.1EAL4+ with CAPP and LSPP / OSPP
- OpenSSL - FIPS 140-2 Level 1 Validated
- CP Assist - SHA-1 validated for FIPS 180-1 - DES & TDES validated for FIPS 46-3

The Common Criteria program establishes an organizational and technical framework to evaluate the trustworthiness of IT Products and protection profiles

- Common Criteria evaluation
  zEC12, zBC12, z13, z13s and LinuxONE servers
  - Common Criteria EAL5+ with specific target of Evaluation – LPAR: Logical partitions
- FIPS
  Crypto Express4s, Crypto Express5s,
  - FIPS 140-2 level 4 Hardware Evaluation
  - Approved by German DK (Deutsche Kreditwirtschaft)

CP Assist
- FIPS 197 (AES)
- FIPS 46-3 (TDES)
- FIPS 180-3 (Secure Hash)

Note:
Common Criteria Certification with Protection Profiles OSPP requires auditing capabilities

© *2017* IBM Corporation

PCI DSS Overview

- The Payment Card Industry Data Security Standard (PCI DSS) was developed to encourage and enhance cardholder data security and facilitate the broad adoption of consistent data security measures globally. PCI DSS provides a baseline of technical and operational requirements designed to protect cardholder data. PCI DSS applies to all entities involved in payment card processing—including merchants, processors, acquirers, issuers, and service providers, as well as all other entities that store, process or transmit cardholder data (CHD) and/or sensitive authentication data (SAD)
- VISA, MasterCard, American Express, …
- PCI DSS comprises a minimum set of requirements for protecting cardholder data, and may be enhanced by additional controls and practices to further mitigate risks, as well as local, regional and sector laws and regulations
- Use of a Payment Application Data Security Standard (PA-DSS) compliant application by itself does not make an entity PCI DSS compliant, since that application must be implemented into a PCI DSS compliant environment

- Existing Assistance: Requirements – with Test Procedures – with Guidance

© 2017 IBM Corporation

PCI DSS requirements

PCI Data Security Standard – High Level Overview

Build and Maintain a Secure Network and Systems	**1.** Install and maintain a firewall configuration to protect cardholder data **2.** Do not use vendor-supplied defaults for system passwords and other security parameters
Protect Cardholder Data	**3.** Protect stored cardholder data **4.** Encrypt transmission of cardholder data across open, public networks
Maintain a Vulnerability Management Program	**5.** Protect all systems against malware and regularly update anti-virus software or programs **6.** Develop and maintain secure systems and applications
Implement Strong Access Control Measures	**7.** Restrict access to cardholder data by business need to know **8.** Identify and authenticate access to system components **9.** Restrict physical access to cardholder data
Regularly Monitor and Test Networks	**10.** Track and monitor all access to network resources and cardholder data **11.** Regularly test security systems and processes
Maintain an Information Security Policy	**12.** Maintain a policy that addresses information security for all personnel

© 2017 IBM Corporation

PCI DSS requirements

PCI Data Security Standard – High Level Overview

Build and Maintain a Secure Network and Systems	1. Install and maintain a firewall configuration to protect cardholder data 2. Do not use vendor-supplied defaults for system passwords and other security parameters
Protect Cardholder Data	3. Protect stored cardholder data 4. Encrypt transmission of cardholder data across open, public networks
Maintain a Vulnerability Management Program	5. Protect all systems against malware and regularly update anti-virus software or programs 6. Develop and maintain secure systems and applications
Implement Strong Access Control Measures	7. Restrict access to cardholder data by business need to know 8. Identify and authenticate access to system components 9. Restrict physical access to cardholder data
Regularly Monitor and Test Networks	10. Track and monitor all access to network resources and cardholder data 11. Regularly test security systems and processes
Maintain an Information Security Policy	12. Maintain a policy that addresses information security for all personnel

© 2017 IBM Corporation

Regularly Monitor and Test Networks

R10: Tack and monitor all access to network resources and cardholder data

- Logging and tracking user activities are critical in preventing, detecting, or minimizing the impact of a data compromise. Logs allow thorough tracking, alerting, and analysis when something does go wrong. Determining the cause of a compromise is very difficult, if not impossible, without system activity logs
  – Application
  – Network
  – System
  – ...
- Linux: Audit Framework, firewall

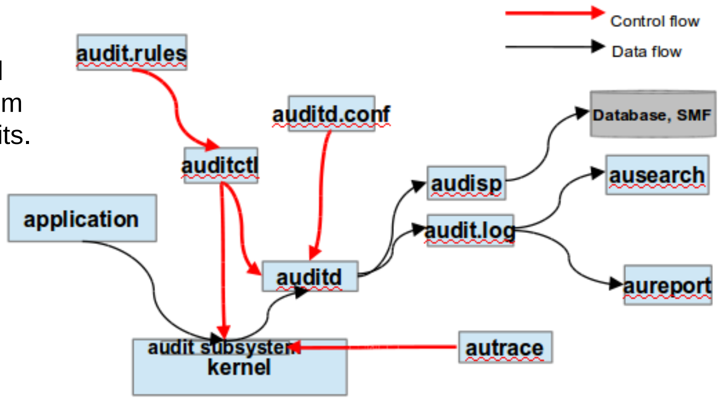
© 2017 IBM Corporation

Regularly Monitor and Test Networks . . .

R10: Tack and monitor all access to network resources and cardholder data

Linux audit Framework

- Collect information regarding events occurring on a running system and helps to make system more secure (Kernel events (system calls) and user events (audit-enbaled programs))
- Form and log a record describing each event using information collected from that event (syscalls arguments, subject attributes, objects attributes, time)
- Analyze the log of records
  – Analyze what is happening
  – However, it does not provide additional security by itself. It does not protect from code malfunctions or any kind of exploits.
  – But it useful for tracking these issues and helps you take additional security measures to prevent the issues from happening or being repeated.



More Infomation

Documentation from RedHat or SUSE

© 2017 IBM Corporation

Protect Cardholder Data

R3: Protect stored cardholder data

- Protection methods such as encryption, truncation, masking, and hashing are critical components of cardholder data protection.
- Crypto: If an intruder circumvents other security controls and gains access to encrypted data, without the proper cryptographic keys, the data is unreadable and unusable to that person.
- Other effective methods of protecting stored data should also be considered as potential risk mitigation opportunities. For example, methods for minimizing risk include not storing cardholder data unless absolutely necessary.

© 2017 IBM Corporation

Protect Cardholder Data . . .

R3: Protect stored cardholder data . . .

- PCI DSS requirements
  - Keep cardholder data storage to a minimum by implementing data retention and disposal policies, procedures and processes . . .
  - Do not store sensitive authentication data after authorization (even if encrypted)
  - Mask PAN (Primary Account Number) when displayed (the first six and last four digits are the maximum number of digits to be displayed) . . .
  - Render PAN unreadable anywhere it is stored (including on portable digital media, backup media, and in logs) . . .
  - Document and implement procedures to protect keys used to secure stored cardholder data against disclosure and misuse
  - Fully document and implement all key-management processes and procedures for cryptographic keys used for encryption of cardholder data

© 2017 IBM Corporation

Protect Cardholder Data . . .

R3: Protect stored cardholder data . . .

- PCI DSS requirements
  - Document and implement procedures to protect keys used to secure stored cardholder data against disclosure and misuse
    - Restrict access
    - Store secret and private keys used to encrypt/decrypt cardholder data in one (or more) of the following forms at all times:
      - Encrypted with a key-encrypting key that is . . .
      - Within a secure cryptographic device such as a host security module (HSM)
      - As at least two full-length key components or key shares, in accordance with an industry- accepted method
    - Store cryptographic keys in the fewest possible locations.
  - Fully document and implement all key-management processes and procedures for cryptographic keys used for encryption of cardholder data
- Linux on z: Consider Secure Key methods (HSM, i.e. CEXnS) for encryption of credit card data (access and management of keys)

© *2017* IBM Corporation

Protect Cardholder Data . . .

R4: Encrypt transmission of cardholder data across open, public networks

- Sensitive information must be encrypted during transmission over networks that are easily accessed by malicious individuals. Misconfigured wireless networks and vulnerabilities in legacy encryption and authentication protocols continue to be targets of malicious individuals
- Use strong cryptography and security protocols (for example, SSL/TLS, IPSEC, SSH, etc.) to safeguard sensitive cardholder data during transmission over open, public networks
- Security policies are defined and in use (use only trusted keys and certificates, encryption strength, never send unprotected PANs by “end-user messaging” technologies (eMail, chat,...)
- Linux for z: ok

© *2017* IBM Corporation

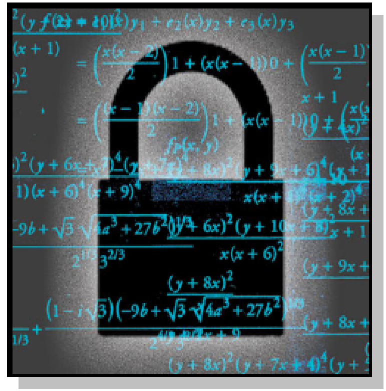
Protection of data: Cryptography with hardware support on Linux for z Systems

Overview, possibilities, and experiences

© 2017 IBM Corporation

Crypto in general: Why?

- Traditionally: to hide the meaning of transferred or stored data, but also used to establish:
  - Data integrity (No alteration)
  - Authentication (Identity Verification)
  - Data confidentiality (Not disclosure)



- A required facility today for personal or industrial computing
- Hardware Cryptography
  - Offload cryptographic computation workload
    - Some algorithms consumes huge amounts of CPU%
  - Increased performance
    - Speed of computation by specialized coprocessors
  - Security
    - Always more secure than a software implementation
    - Can implement very sophisticated protection of secrets, depending on device

© 2017 IBM Corporation

Crypto in general: Algorithms and their usage

Authentication, key distribution
(VPN, SSL/TLS handshake, ..)

ECC
RSA (512, 1024, 2048, 4096 bits)
DSA (512, 1024 bits)

Private
Public

DES (56 bits)
T-DES (168 bits)
AES (128 bits)
AES (192, 256 bits)

Data transfer (VPN, SSL/TLS, ..)
Data storage (database, archives, ...)

\> **Symmetric algorithms (Shared Secret Key)**
Confidentiality of data

\> **Asymmetric algorithms (Public Key Cryptography)**
Digital signature
Confidentiality of key

\> **Cryptographic "checksum" with one way algorithms**
Integrity of data

MD5 (128-bit hash)
MAC, MDC
SHA-1(160-bit hash)
SHA-256 (256-bit hash)
SHA-384 , SHA-512

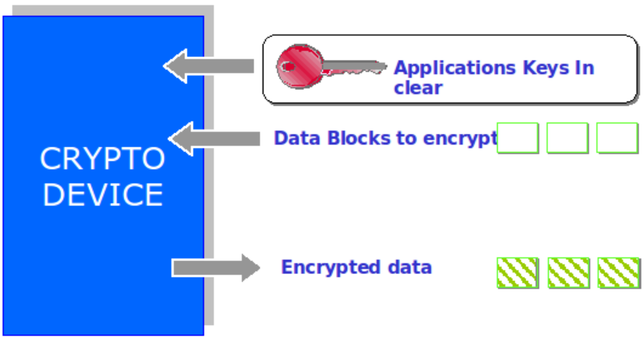
Data transfer (VPN, SSL/TLS, ..)
Data storage (database, archives, ...)

© *2017* IBM Corporation

Crypto in general: Clear Key implementation

"Clear Key – key may be in the clear, at least briefly, somewhere in the environment"

CPACF, CEX5A



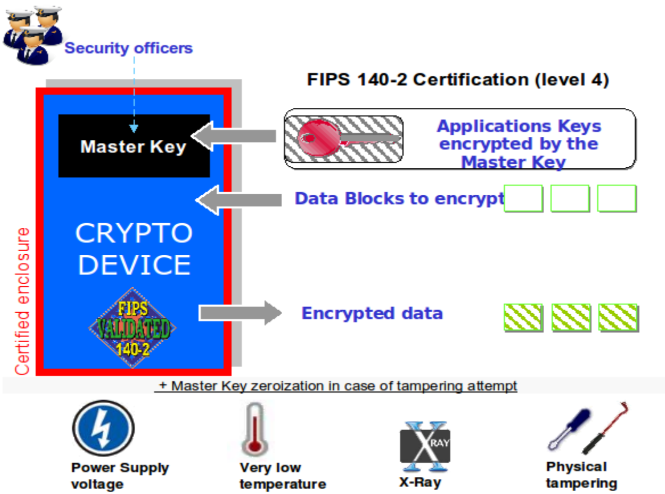
© 2017 IBM Corporation

Crypto in general: Secure Key implementation

- Secure Coprocessor - HSM

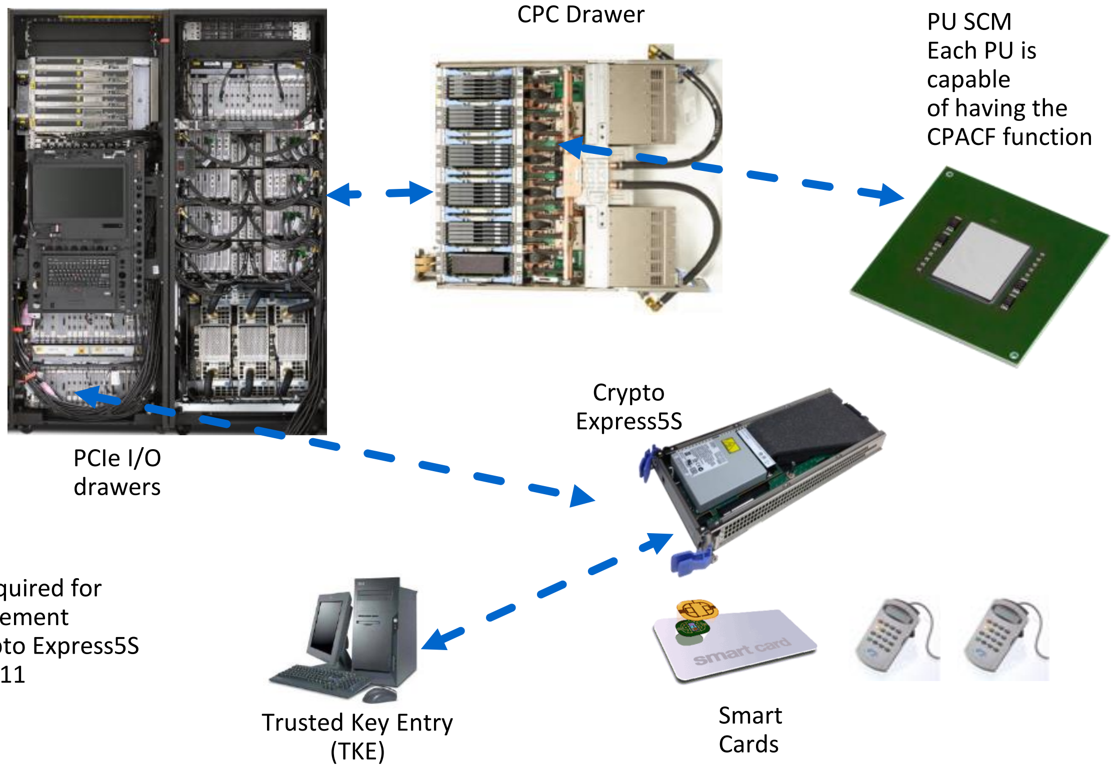
"Secure Key – key value does not exist in the clear outside of the HSM (secure, tamper-resistant boundary of the card)"

CEX5C, CEX5P
CEX6C, CEX6P



© 2017 IBM Corporation

Crypto in general:
HW Crypto support in IBM Z (here z13)

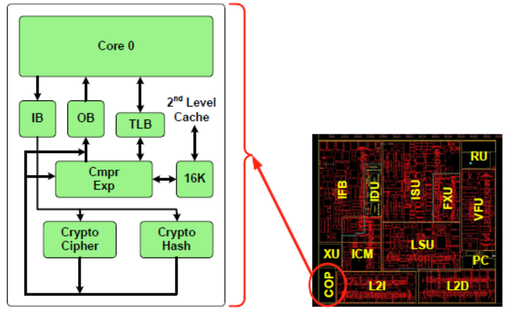


TKE required for
management
of Crypto Express5S
and EP11

© 2017 IBM Corporation

CPACF - CP Assist for Cryptographic Functions

- Available on every Processor Unit defined as a CP, IFL, zAAP and zIIP
- non-privileged instructions supporting:
  - hashes/MACs: SHA1, SHA 2 (224,256, 384, 512), SHA3 (224, 356, 384, 512), SHAKE (128, 256), GHASH,
  - symmetric ciphers: DES, 2DES, 3DES, AES-128, AES-192, AES-256
  - modes of operations: ECB, CBC, CTR, OFB, CFB, XTS, CBC-MAC, GCM, (CMAC, CCM)
  - pseudo random number generation: 3DES based PRNG, NIST SP-800-90A SHA-512 based DRNG
  - true random number generation
- Must be explicitly enabled, using a no-charge enablement feature (#3863),
  - SHA algorithms are always available
- Protected key support for additional security of cryptographic keys



Note:
Performance improvement for CPACF on z13 vs z12 and on z14 vs. z13

© 2017 IBM Corporation

Crypto Express Adapters

- Adapter virtualization

Adapter can be partitioned into different domains (separate master keys per doamin)

<= CEX4S: 16 domains

CEX5S, CEX6S: 85 domains

- CCA: Classical IBM standard
- PKCS11: Industry Standard (distr.)

- **Three configuration options for the PCIe adapter**
  - Only one configuration option can be chosen at any given time
  - Switching between configuration modes will erase all card secrets (Exception: Switching from CCA to accelerator or vice versa)

- **Accelerator**
  - For SSL acceleration
  - Clear key RSA operations

- **Enhanced: Secure IBM CCA coprocessor (default)**
  - Optional: TKE workstation (FC 0841) for security-rich, flexible key entry or remote key management

- **IBM Enterprise PKCS #11 (EP11) coprocessor**
  - Designed for extended evaluations to meet public sector requirements
    - Both FIPS and Common Criteria certifications
  - Required: TKE workstation (FC 0841) for management of the Crypto Express4S when defined as an EP11 coprocessor

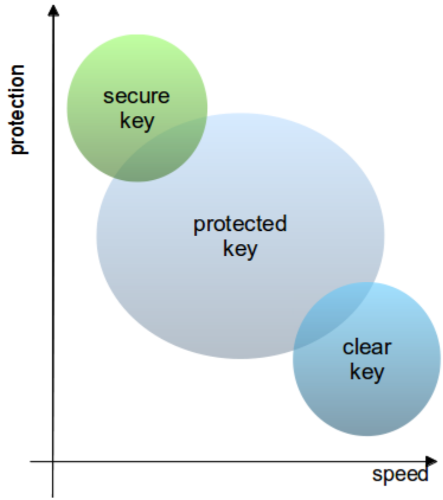
© 2017 IBM Corporation

3 levels of protection – 3 levels of speed

- **Clear Key** – key is in the clear, at least briefly, somewhere in the environment
  - Example use: SSL transaction security

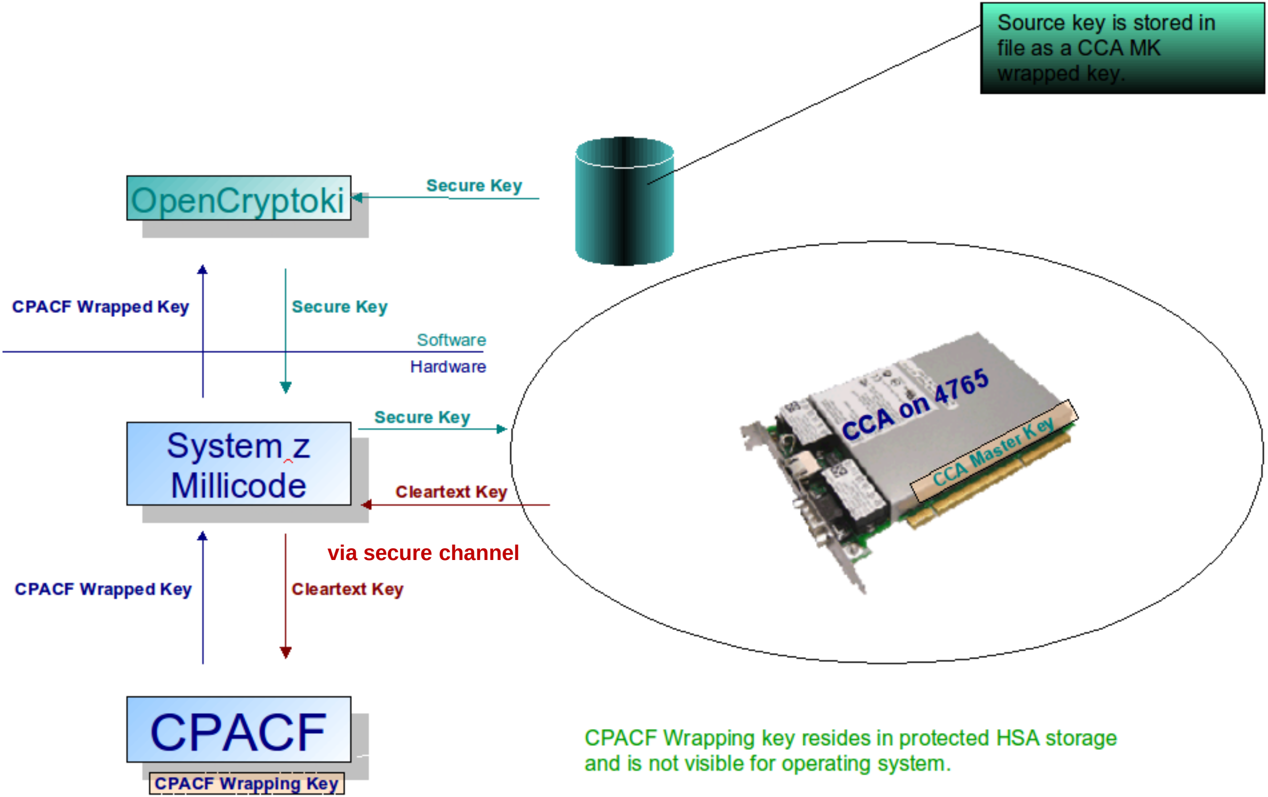
- **Protected Key** – key value does not exist outside of physical hardware, although the hardware may not be tamper-resistant
- Unique to z Systems
  - Example use: protection of data at rest
  - csu_hcpuaprt has to be set

- **Secure Key** – key value does not exist in the clear outside of the HSM (secure, tamper-resistant boundary of the card)
  - Example use: PIN handling and verification



© 2017 IBM Corporation

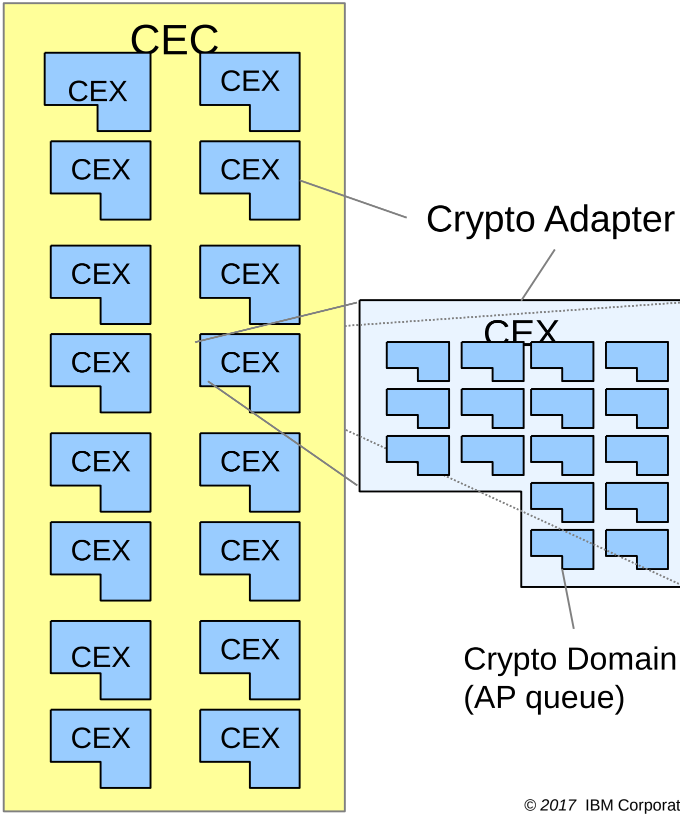
Secure Key CPACF - Key Wrapping



CPACF Wrapping key resides in protected HSA storage and is not visible for operating system.

On Features, Adapters, APs, Domains, Queues. . . .

- CEX5 feature has 1 adapter (aka AP). Up to 16 CEX5 features per CEC
- Each adapter has an AP Id
- Each adapter has a mode
  - coprocessor CCA or EP11 mode, or
  - accelerator
- Each adapter can be divided in up to 85 domains (HW virtualization)
- each domain in an AP is represented in SW by an AP queue
- Each domain can hold domain secret (master key)
- Configuration constraints
  - each LPAR may be granted access to
    - a list (a_1, a_2, ..., a_k) of APs and
    - a list (d_1,d_2, … d_j) of domains
  - resulting in access to AP queues
    - (a_1d_1, ..., a_1d_j, a_2d_1, ..., a_kd_j)
- The Linux on z device driver (until kernel 4.9)
  - only uses one domain/AP queue per AP

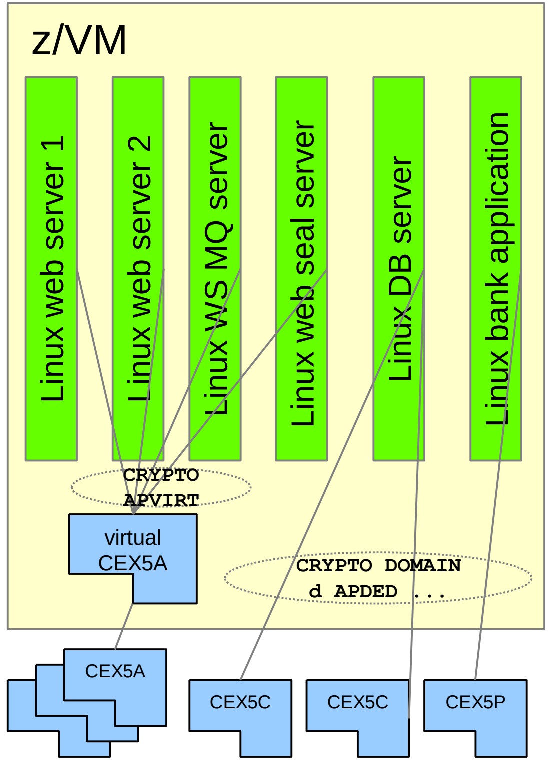


© 2017 IBM Corporation

z/VM Crypto Guest Support

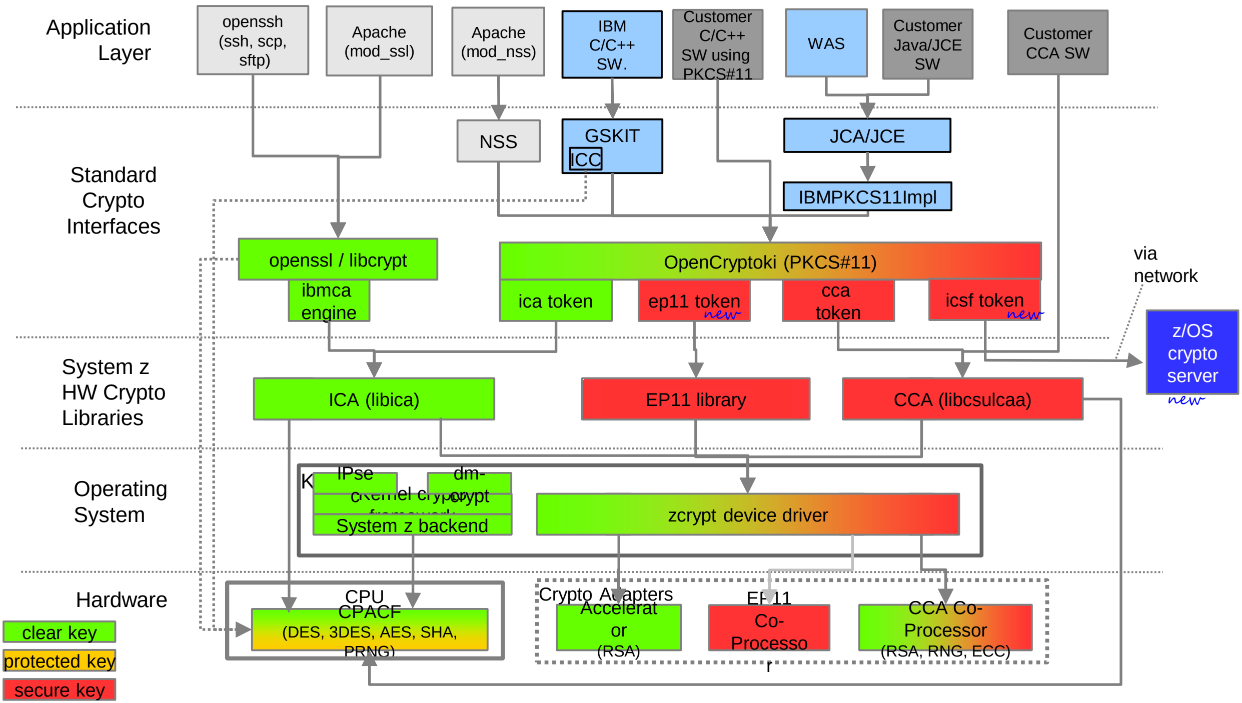
- A guest may have
  - either dedicated adapters
    - `CRYPTO DOMAIN d APDED a1 a2 ...`
  - or shared adapters
    - `CRYPTO APVIRT`
- Shared adapters
  - are of a single type
    - uses only highest priority type
    - priority:
      CEX6A > CEX5A > CEX4A >... > CEX52C >CEX4C >...
  - clear key operations
- Checking Crypto Configuration
  - show status of crypto facilities
    - `Q CRYPTO [ DOMAINs [Users] ]`
  - show status of crypto facilities of guest
    - `Q V CRYPTO`

Careful Planning is required!

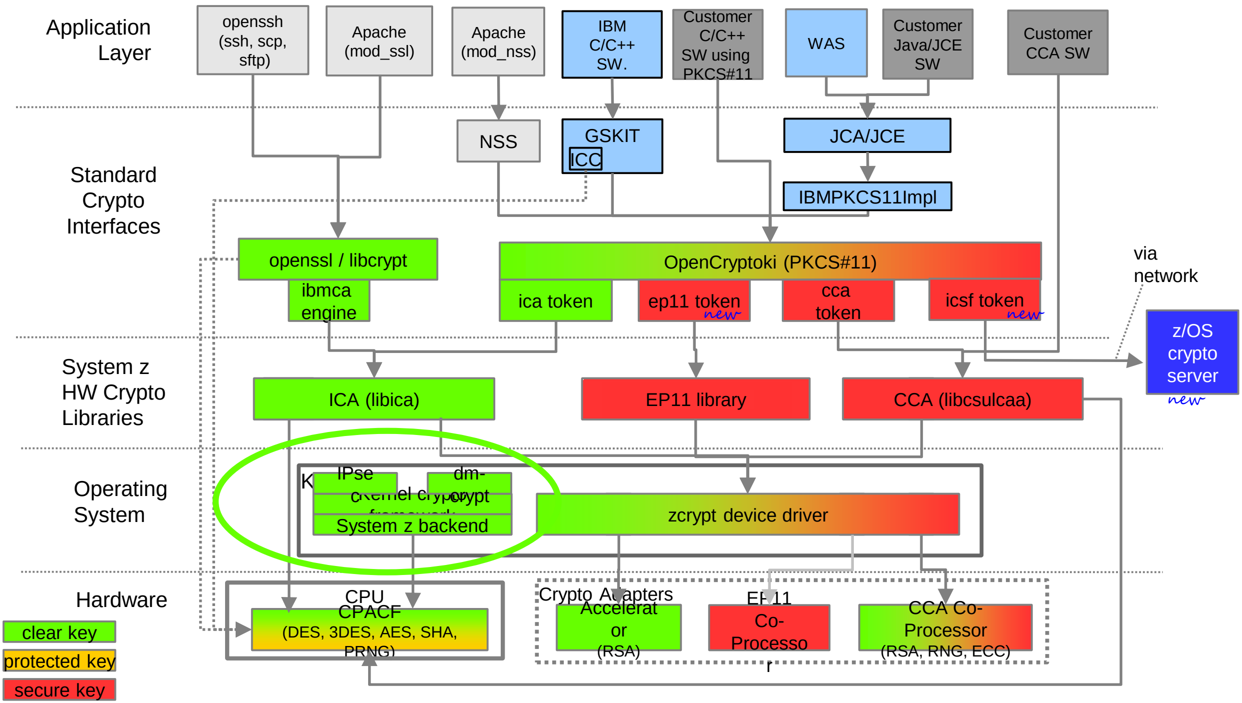


© 2017 IBM Corporation

Crypto in general: Linux on z Crypto Stack

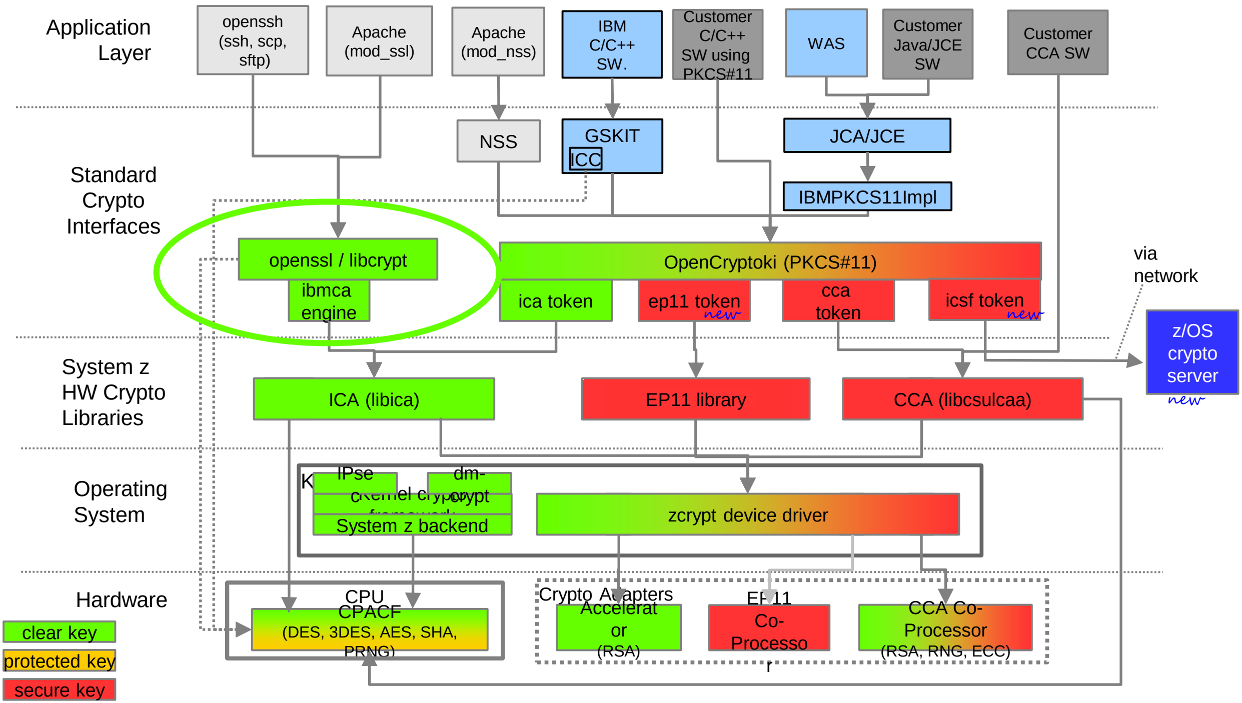


Crypto in general: Linux on z Crypto Stack

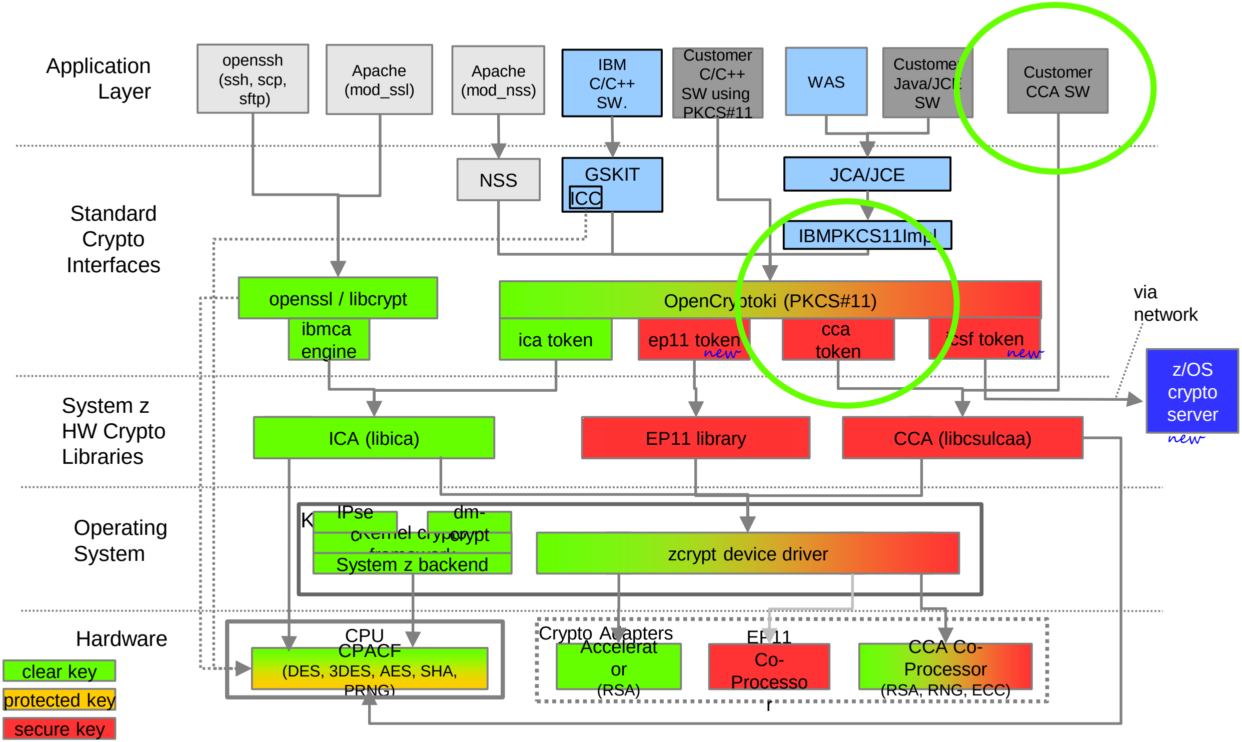


© 2017 IBM Corporation

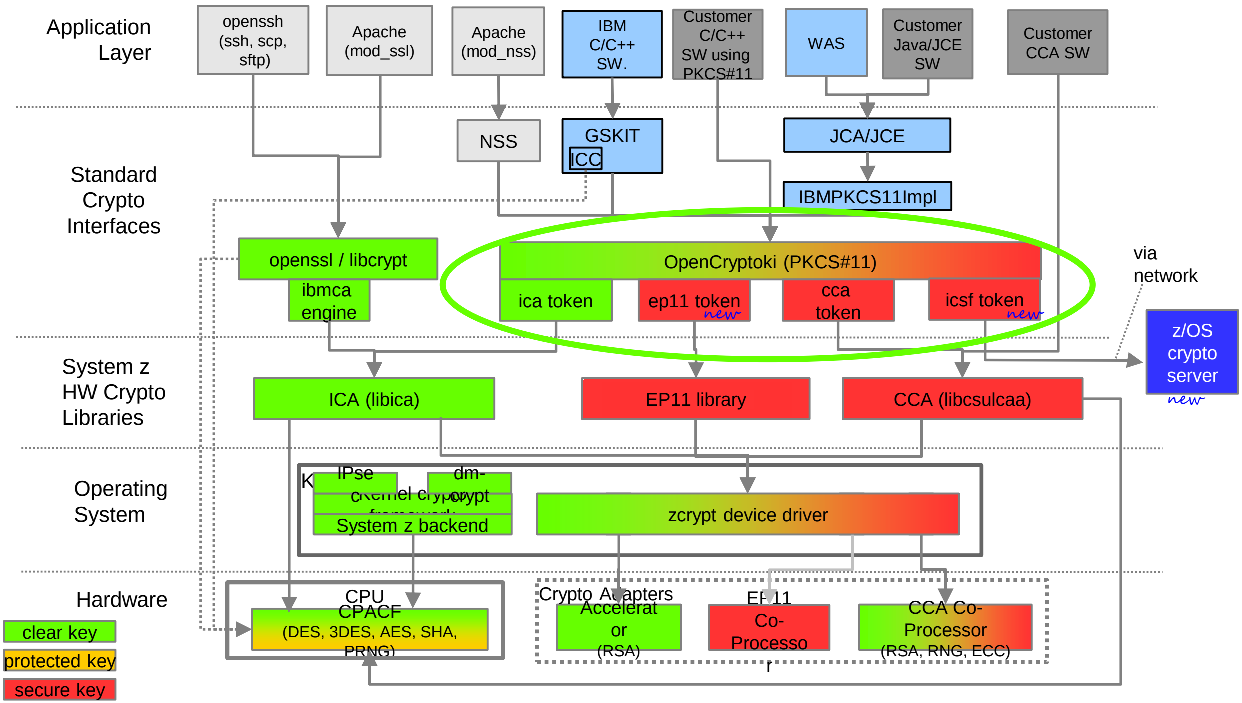
Crypto in general: Linux on z Crypto Stack



Crypto in general: Linux on z Crypto Stack



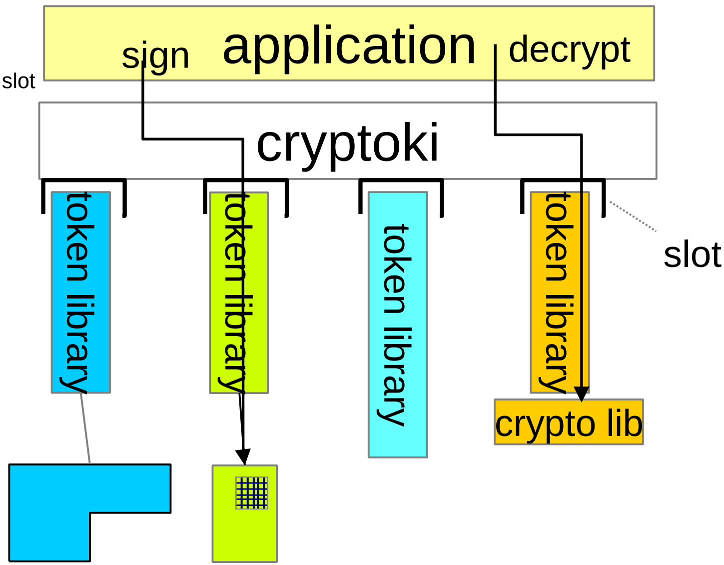
Crypto in general: Linux on z Crypto Stack



© 2017 IBM Corporation

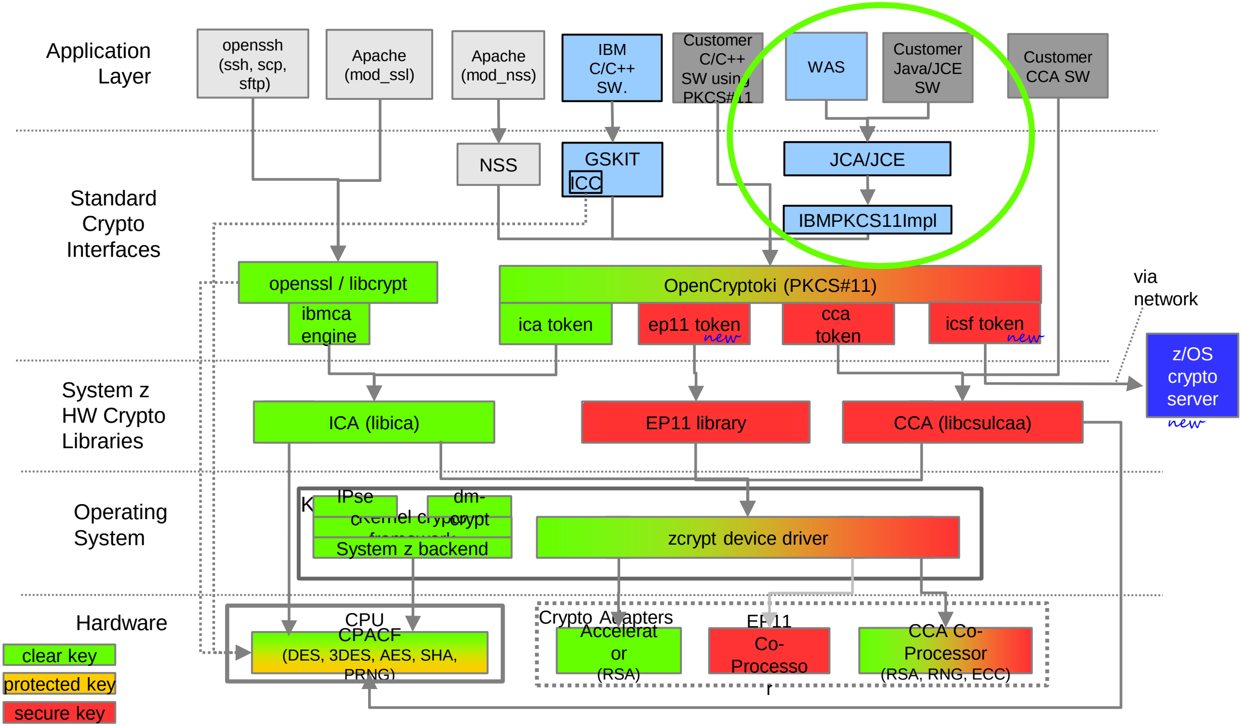
PKCS#11 Concepts: Slots and Tokens

- Model: smart cards and readers
  - reader: slot
  - crypto processor: token to be inserted in slot
- slots and tokens may be HW specific
- slot and token functions
  - C_GetSlotList(), C_GetSlotInfo(),
  - C_WaitForSlotEvent()
  - C_InitToken(), C_GetTokenInfo()
  - C_initPIN(), C_SetPIN()
- slot info
  - token present
  - device removable
  - ...
- token info
  - login required,
  - too many wrong pins entered
  - has RNG
  - ...



© 2017 IBM Corporation

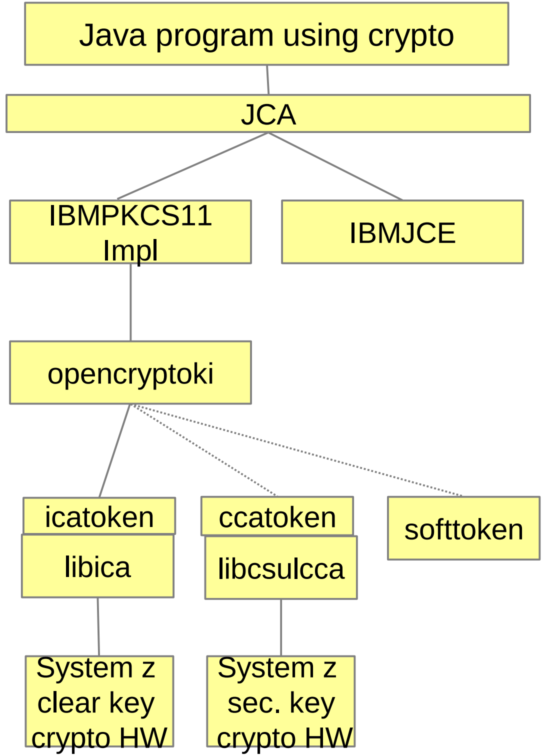
Crypto in general: Linux on z Crypto Stack



© 2017 IBM Corporation

PKCS#11 und Java
The Java Cryptographic Architecture (JCA)

- Java Cryptographic Architecture (JCA)
  - provider architecture for security APIs
  - supports multiple providers with different priorities and capabilities
  - providers that implement the JCE API:
    - IBMJCE (software implementation by IBM equivalent to SunJCE)
    - IBMPKCS11Impl calls openCryptoki which can be configured to use a specific token to exploit crypto HW support
      - clear key crypto via libica
      - secure key crypto via CCA library
- Java Cryptographic Extension (JCE)
  - API for basic cryptographic functions



© 2017 IBM Corporation

Configuring Java for HW Crypto Usage

The `java.security` file maintains a list of available JCA providers

standard location:

`/usr/lib/jvm/java-`<version>`-ibm-`<ext. version>`.s390x/jre/lib/security/java.security`

Example extract from java.security

```
...
# List of providers and their preference orders (see above):
#
security.provider.1=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl /root/zpkcs.cfg
security.provider.2=com.ibm.crypto.provider.IBMJCE
#security.provider.3=com.ibm.security.jgss.IBMJGSSProvider
...
```

The IBMPKCS11Impl has a configuration file as argument

Example configuration file for IBMPKCS11Impl:

```
name = Sample
description = Sample config for z/linux
library = /usr/lib64/pkcs11/PKCS11_API.so
# the following references the icatoken
slot = 0
# the following references the ccatoken
#slot = 1
# the following references the softtoken
#slot = 2
disabledmechanisms = { CKM_SHA_1 }
```

Atention:
Syntax is very “faible”

© 2017 IBM Corporation

z Systems Architecture Support for Java 8

- The IBM Java 8 Java Cryptography Extension (JCE) on z Systems
  - uses CPACF instructions to accelerate
    - DES, 3DES, AES
      - with ECB, CBC. OFB, CFB and CFB x modes of operation
    - SHA1 and SHA2
  - uses z Systems specific code to accelerate
    - ECDHE – NIST P256 and ECDHE-ECDSA

© 2017 IBM Corporation

PKCS#11 and Standard SW

standard middleware often provides for a plug-in option for PKCS#11 libraries

- IBM WebSphere Application Server (WAS) via Java
- Also other Application Server via Java
- IBM HTTP Server (IHS) via GSKIT
- NSS

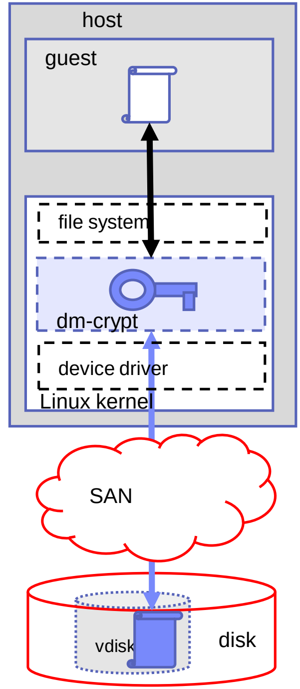
configuration files of such software may allow to specify

- library path of opencryptoki
- slot or token id
- user PIN

© 2017 IBM Corporation

Using dm-crypt for Guest Data Encryption

- **dm-crypt / LUKS**
  - a mechanism for end-to-end data encryption
  - data only appears in the clear when in program
- kernel component that transparently
  - for a whole block device (partition or LV)
    - encrypts all data written to disk
    - decrypts all data read from disk
- How it works:
  - encryption keys stored on disk (partition, LV)
  - encryption keys on disk are protected by passwords
  - uses in kernel-crypto
    - can use HW crypto
      - Linux on z has HW support for
        - > AES-CBC
        - > XTS-AES (recommended)



What are Containers

- Virtual environment within Linux OS instance
  - So applications share OS kernel
  - Only application is started, not entire Linux environment
- Efficiency: no virtualization overhead
  - No full system or para-virtualization, but isolation only by the kernel
- Own file system tree via chroot environment
- Container separation of OS objects via „name spaces“
  - Process IDs, network devices, mount points, users, and more

© 2017 IBM Corporation

Docker: „Build, Ship, and Run Any App, Anywhere“

- One implementation of a container solution
- Powerful tool to build, modify, deploy, run, manage containers
  - Extreme focus on efficiency, fast response times
  - Stores incremental differences and caching whenever possible
- Registries serve as central places for images
  - Efficient distribution, versioning
- Terminology
  - image: a self contained set of files, base for a container
  - container: runnable instance, based on an image
- Maintained by Docker, Inc.

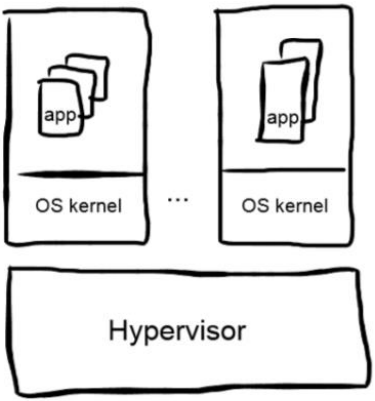
© 2017 IBM Corporation

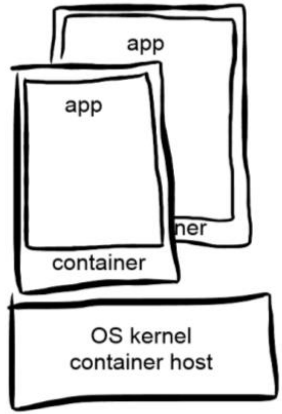
Typical Container Attributes

- Self contained sets of files – escape dependency hell, reduce test matrix
- Serve a single task
- Can build on top of each other
- Can be deployed simple and quickly
- Can easily be customized, re-packaged and versioned
- Can use synergies in the kernel, if images eventually base on the same libraries (same file in underlying images)
  - without having to use KSM (Kernel Samepage Merging)

© *2017* IBM Corporation

Virtualization vs. Containers





Infrastructure oriented:

- coming from servers, now virtualized
- virtual server resource management
- several applications per server
- isolation
- persistence

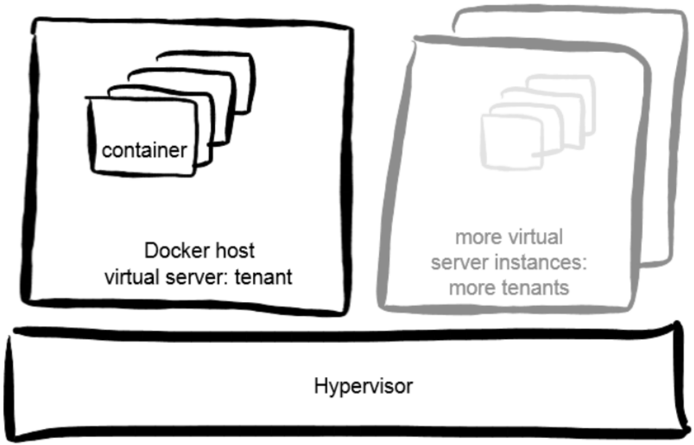
Service oriented:

- application-centric
- application management
- solution decomposed
- DevOps
- dynamic

© 2017 IBM Corporation

Virtualization and Containers

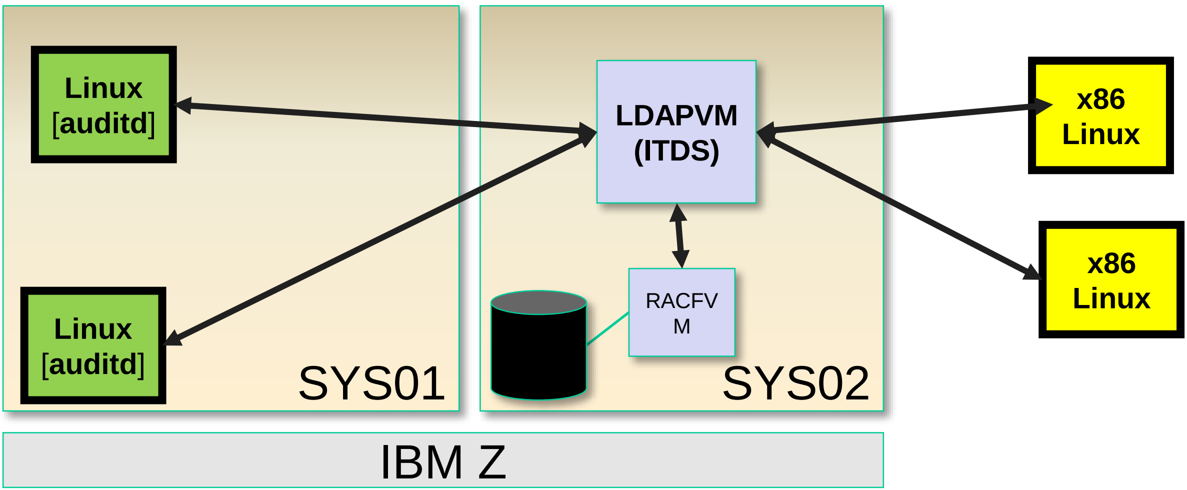
- Virtual machine separation between tenants
  - Virtualization management for infrastructure
  - Isolation
- Many containers within tenants
  - Container efficiency
  - Docker management and ecosystem



© 2017 IBM Corporation

Centralized Audit with ITDS and RACF

- Common client (auditd with appropriate PAM plug-ins)
- Integrated LDAP server (ITDS, same on z/VM as on z/OS)
- LDAP uses RACFVM as a DBM back-end.

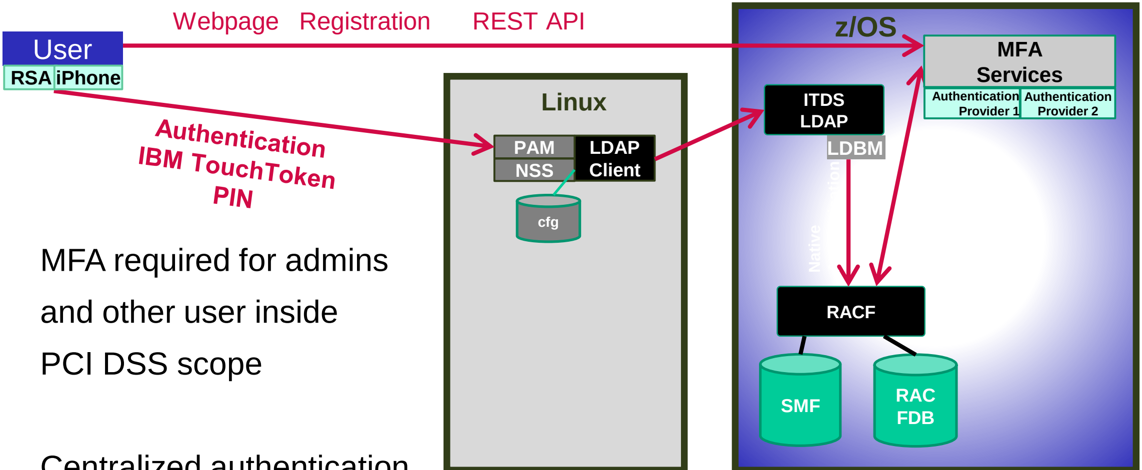


© 2017 IBM Corporation

Centralized authentication and MFA

Requirements for PCI DSS 3.2: MFA

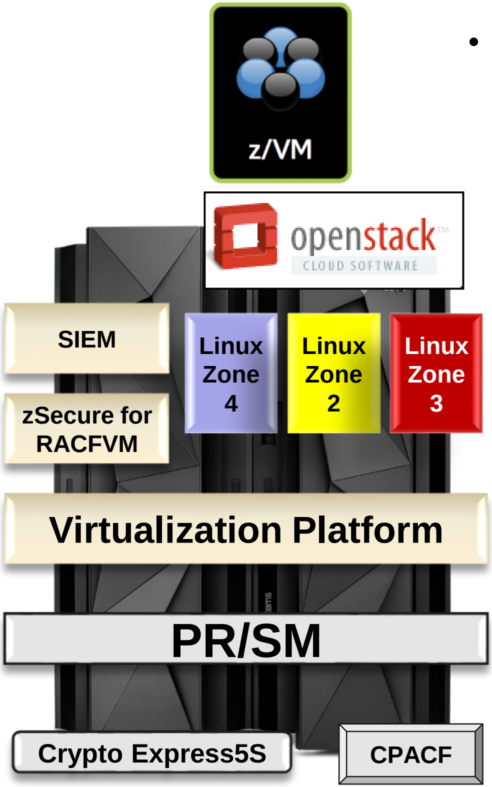
Detailed infos about experiences will come soon . . .



MFA required for admins and other user inside PCI DSS scope

Centralized authentication using ITDS / RACF: central user management

Summary:
Linux on z provides ultimate security at scale.



- ***z/VM** represents **40+** years of virtualization security*

IaaS on z Systems for Linux
OpenStack for compatibility and open standards
Keystone for Identity Management and Integration

Linux Security (SELinux, AppArmor, cgroups)
OpenSSH for secure guest connectivity
Centralized Audit with PAM and ITDS

Architecture-layer guest isolation
TLS 1.2 connectivity & VLAN-aware Virtual Switch
OSPP EAL 4+ with Labeled Security (Multitenancy)

Architecture-layer isolation of workload
Ultimate partition isolation (CC EAL 5)
Hipersockets for secured internal traffic

Hardware acceleration of cryptographic ops
PKCS #11 and CCA support
FIPS 140-2 Level 4 HSM (Secure Key)

© 2017 IBM Corporation

Herzlichen Dank

IBM

Dr. Manfred Gnirss

IBM Client Center –
Systems and Software –
z ATS

IBM Germany Lab

Schoenaicher Str. 220
D-71032 Boeblingen
Phone +49 (0) 7031 16-4093
gnirss@de.ibm.com

IBM **Client Center**
Boeblingen

Engage with technical experts and thought leaders from IBM Germany Research and Development in the IBM Client Center in Boeblingen. Discuss and exchange ideas about the latest IBM solutions and technologies. The Boeblingen Lab has a unique set of skills to help you in focus areas like IBM Systems, Cloud, Internet of Things (IoT), Analytics, Mobile, Security, and Commerce.

What we offer

- **Innovation Workshops and IBM Design Thinking:** Share ideas and co-create value
- **Proof of Concept Studies:** Demonstrate the feasibility of solutions in your IT environment
- **Support for Business Partners and Independent Software Vendors (ISVs):** Leverage our expertise to expand your business
- **Briefings:** Solution and technology briefings tailored to your specific needs
- **Demos and Showcases:** Experience IBM solutions live - across IT systems and software
- **Lab Advocacies - Leverage the Lab:** Build relationship with experts from our development team

Contact us

IBM Client Center - IBM Germany Research & Development GmbH
Schoenaicher Str. 220
D - 71032 Boeblingen
mailto:clientcenter@de.ibm.com
http://www.ibm.com/ibm/clientcenter/boeblingen

© 2016 IBM Corporation

Misc.

© 2017 IBM Corporation

References

Hardware cryptographic support wuth Ubuntu Server for IBM Linux for z Systems abd LinuxONE, by Klaus Bergmann, Reinhard Buendgen, Uwe Denneler, Jonathan Furminger, Frank Heimes, Manfred Gnirss, Christian Rund, Patrick Steuer, Arwed Tschoeke.
https://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102721

Hardware cryptographic support of IBM z Systems for OpenSSH in RHEL 7.2 and SLES 12 SP1, by Uwe Denneler, Harald Freudenberger,
Paul Gallagher, Manfred Gnirss, Guillaume Hoareau, Arwed Tschoeke, Ingo Tuchscherer, Arthur Winterling.
https://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102653

First experiences with hardware cryptographic support for OpenSSH with Linux for z Systems, by Manfred Gnirss, Winfried Mueunch, Klaus Werner, and Arthur Winterling.
http://ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101690

Using Crypto Hardware With Java in Linux on System z, by Reinhard Buendgen, Peter Spera in Enterprise Tech Journal on March 20, 2013.
http://enterprisesystemsmedia.com/article/using-crypto-hardware-with-java-in-linux-on-system-z#sr=g&m=o&cp=or&ct=-tmc&st=(opu%20qspwjefe)&ts=1485100746

Using Linux on System z Hardware Cryptography With the PKCS#11 Cryoptography Stack, by Reinhard Buendgen in Enterprise Tech Journal on October 6, 2014.
http://enterprisesystemsmedia.com/article/using-linux-on-system-z-hardware-cryptography-with-the-pkcs11-cryoptography#sr=g&m=o&cp=or&ct=-tmc&st=(opu%20qspwjefe)&ts=1485100746

Configuring an Apache mod_nss server to exploit z Systems cryptographic hardware, by Patrick Steuer, Reinhard Buendgen, George C. Wilson.
http://www.ibm.com/support/knowledgecenter/linuxonibm/liaag/wnsf/l0wnsf00_2015.htm

© *2017* IBM Corporation