TRESCCA

SEVENTH FRAMEWORK PROGRAMME

Project acronym:	**TRESCCA**	
Project title:	TRustworthy Embedded systems for Secure Cloud Computing	
Project number:	European Commission – 318036	
Call identifier:	FP7-ICT-2011.1.4	
Start date of project:	01 Oct. 2012	**Duration:** 36 months

Document reference number:	**D5.6**
Document title:	Report on standardization and open-source activities
Version:	1.0
Due date of document:	September 2015
Actual submission date:	September 2015
Lead beneficiary:	VOSYS
Participants:	Michele Paolino, Ignacio Garcia Vega (WT), Bernhard Katzmarski (OFFIS), Miltos Grammatikakis (TEI), Guillame Duc (TP)
Reviewers:	//

Project co-funded by the European Commission within the 7th Framework Programme		
DISSEMINATION LEVEL		
PU	Public	X
PCA	Public with confidential annex	
CO	Confidential, only for members of the consortium (including Commission Services)	

CONTENTS

1 INTRODUCTION

1.1 Purpose of the Document

This document is the deliverable D5.6 of the TRESCCA project. It aims to describe the activities performed by TRESCCA partners to open and disseminate the project's technologies in regard to open source communities, standardization initiatives and other projects funded by the European Commission.

1.2 Document Versions Sheet

Version	Date	Description, modifications, authors
2.0	2015-09-22	Initial version.
1.0	2015-08-10	First draft.

2 TRESCCA OPEN SOURCE, STANDARDIZATION AND PROJECT COOPERATION ACTIVITIES

Open source projects and standardization activities are gaining more and more consensus in recent years e.g., the Linux kernel, Android and Open Stack but also Networking Function Virtualization (NFV), Open Computing Language (OpenCL), etc. As a consequence, the number of open source users is increasing along with their awareness of the benefits that open solutions bring to the every day digital life.

However, the impact that open source has on security matters is even more important. In fact, the possibility to publicly access code sources and standard descriptions is, for security researchers, of pivotal importance to find security breaches and improve the security of the hardware and software. Projects such as Open Secure Sockets Layer (OpenSSL) and Open Secure Shell (OpenSSH) are benefitting from this openness since years, in the same way as open standards gain more users when the standard is easily accessible, e.g., GlobalPlatform Trusted Execution Environment (TEE).

One of the main objectives of TRESCCA is to improve the trustworthiness of future cloud services. This requires wide adoption of the developed technology as well as the establishment of non-proprietary and open standards. For this purpose, the TRESCCA consortium created an open-access model for the project results by the following activities:

- Release of the TRESCCA security extensions under the open-source license.
- Submission of patches and modules to existing open sources projects such as QEMU or the Linux kernel.
- Public release of SW and HW-API definitions and functional specification of HW security module.
- Disseminate the TRESCCA experience and results with other European Commission projects.

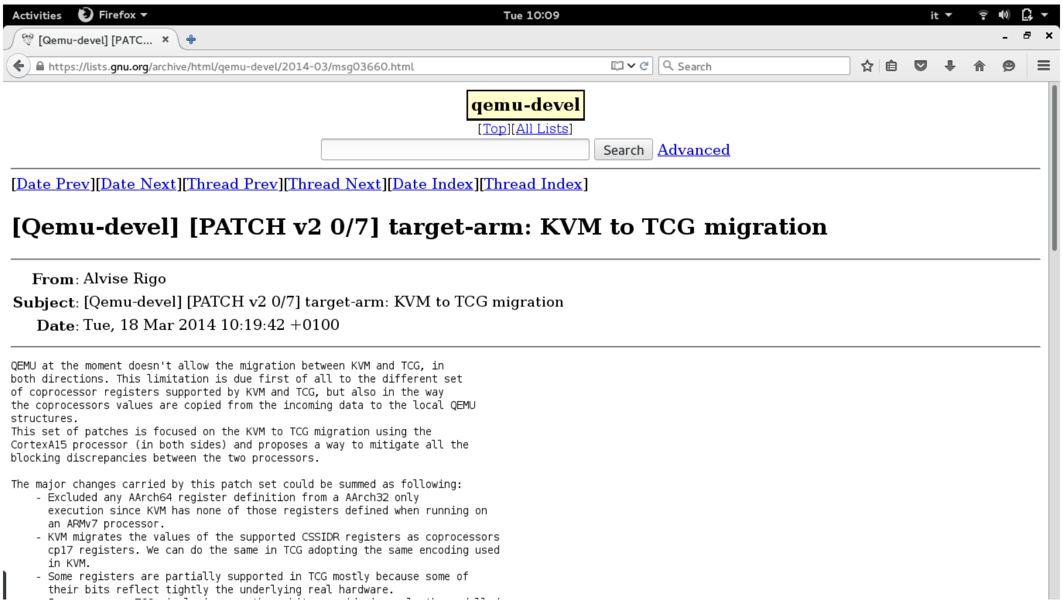
In the following section, the standardization, open source and project cooperation activities for each partner will be detailed.

2.1 Contributions

2.1.1 Virtual Open Systems

During the TRESCCA activity, Virtual Open Systems extended various open sources projects, adding functions or features which belong to the TRESCCA Secure hypervisor. The source code of these contributions is publicly available in the related mailing lists (Figure 2.1 and 2.2), but can be also downloaded as source code from the Virtual Open Systems' git server (git.virtualopensystems.com).

- **QEMU** Quick-Emulator (QEMU) is a key component of the KVM hypervisor, because it emulates the machine models used to run virtual machines. Thanks to TRESCCA, VOSYS submitted two patches to the QEMU community:



Activities
Firefox
Tue 10:09
it
[Qemu-devel] [PATC...
https://lists.gnu.org/archive/html/qemu-devel/2014-03/msg03660.html
Search
qemu-devel
[Top][All Lists]
Search
Advanced
[Date Prev][Date Next][Thread Prev][Thread Next][Date Index][Thread Index]

[Qemu-devel] [PATCH v2 0/7] target-arm: KVM to TCG migration

From: Alvise Rigo
Subject: [Qemu-devel] [PATCH v2 0/7] target-arm: KVM to TCG migration
Date: Tue, 18 Mar 2014 10:19:42 +0100

```
QEMU at the moment doesn't allow the migration between KVM and TCG, in
both directions. This limitation is due first of all to the different set
of coprocessor registers supported by KVM and TCG, but also in the way
the coprocessors values are copied from the incoming data to the local QEMU
structures.
This set of patches is focused on the KVM to TCG migration using the
CortexA15 processor (in both sides) and proposes a way to mitigate all the
blocking discrepancies between the two processors.

The major changes carried by this patch set could be summed as following:
    - Excluded any AArch64 register definition from a AArch32 only
      execution since KVM has none of those registers defined when running on
      an ARMv7 processor.
    - KVM migrates the values of the supported CSSIDR registers as coprocessors
      cp17 registers. We can do the same in TCG adopting the same encoding used
      in KVM.
    - Some registers are partially supported in TCG mostly because some of
      their bits reflect tightly the underlying real hardware.
```

Figure 2.1: QEMU hybrid migration patches

- AARCH64 support on machvirt machine model using KVM[1]: Implements the KVM support in QEMU for the ARMv8 Cortex A57 CPU. This implementation supports both 64-bit and 32-bit guests on AARCH64.
- KVM to TCG hybrid migration[2]: This patch enables the migration between KVM and TCG for CortexA15, in both directions. This patch series enables hybrid migration by overcoming the QEMU limitations due to different set of coprocessor registers supported by KVM and TCG, but also to different techniques used to copy coprocessors values from the incoming data to the local QEMU structures.

The company git repo which contains these contributions is: git@git.virtualopensystems.com:trescca/qemu.git

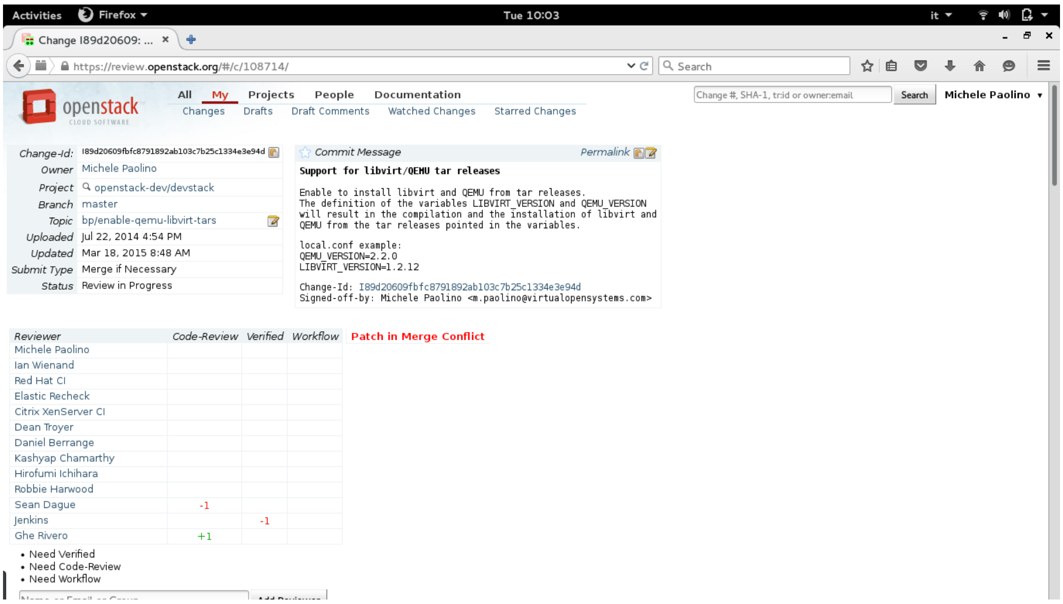
- **devstack**: Devstack is set of scripts and utilities used to quickly deploy an OpenStack cloud. It is widely used in Continuous Integration systems and has a very active community. VOSYS developed an extension for devstack - Enable to install libvirt and QEMU from tar releases[3] - which enables to deploy a full cloud stack with TRESCCA libvirt and qemu versions. The patch has been shared with the OpenStack community (Figure 2.2), and received many positive comments. The code of this work is also publicly available at git.virtualopensystems.com:trescca/devstack.git

2.1.2 OFFIS

OFFIS contribution related to open source activity consists of some repositories available on the TRESCCA github account [4].

As shown in Figure 2.3, the git repository contains the following components:

[1] https://lists.cs.columbia.edu/pipermail/kvmarm/2013-September/005779.html
[2] https://lists.gnu.org/archive/html/qemu-devel/2014-03/msg03660.html
[3] https://review.openstack.org/#/c/108714/
[4] github.com/TRESCCA



Change-Id:	I89d20609fbfc8791892ab103c7b25c1334e3e94d
Owner	Michele Paolino
Project	openstack-dev/devstack
Branch	master
Topic	bp/enable-qemu-libvirt-tars
Uploaded	Jul 22, 2014 4:54 PM
Updated	Mar 18, 2015 8:48 AM
Submit Type	Merge if Necessary
Status	Review in Progress

Commit Message

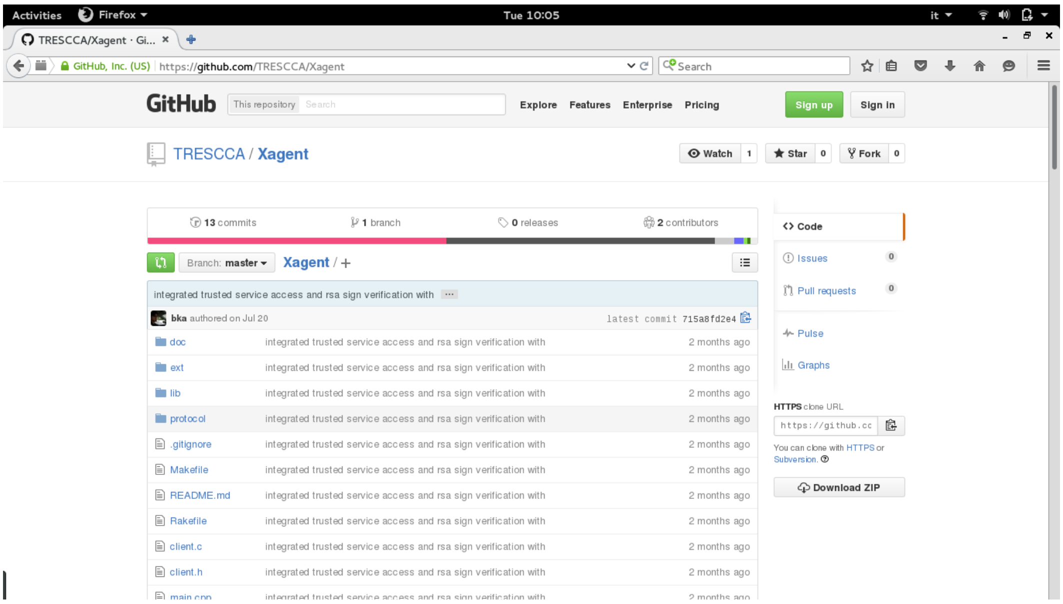
```
Support for libvirt/QEMU tar releases

Enable to install libvirt and QEMU from tar releases.
The definition of the variables LIBVIRT_VERSION and QEMU_VERSION
will result in the compilation and the installation of libvirt and
QEMU from the tar releases pointed in the variables.

local.conf example:
QEMU_VERSION=2.2.0
LIBVIRT_VERSION=1.2.12

Change-Id: I89d20609fbfc8791892ab103c7b25c1334e3e94d
Signed-off-by: Michele Paolino <m.paolino@virtualopensystems.com>
```

Reviewer	Code-Review	Verified	Workflow
Michele Paolino			
Ian Wienand			
Red Hat CI			
Elastic Recheck			
Citrix XenServer CI			
Dean Troyer			
Daniel Berrange			
Kashyap Chamarthy			
Hirofumi Ichihara			
Robbie Harwood			
Sean Dague	-1		
Jenkins		-1	
Ghe Rivero	+1		

Patch in Merge Conflict

- Need Verified
- Need Code-Review
- Need Workflow

Figure 2.2: The devstack patches developed by VOSYS



TRESCCA / Xagent

13 commits · 1 branch · 0 releases · 2 contributors

Branch: master · Xagent / +

integrated trusted service access and rsa sign verification with …

bka authored on Jul 20 · latest commit 715a8fd2e4

doc	integrated trusted service access and rsa sign verification with	2 months ago
ext	integrated trusted service access and rsa sign verification with	2 months ago
lib	integrated trusted service access and rsa sign verification with	2 months ago
protocol	integrated trusted service access and rsa sign verification with	2 months ago
.gitignore	integrated trusted service access and rsa sign verification with	2 months ago
Makefile	integrated trusted service access and rsa sign verification with	2 months ago
README.md	integrated trusted service access and rsa sign verification with	2 months ago
Rakefile	integrated trusted service access and rsa sign verification with	2 months ago
client.c	integrated trusted service access and rsa sign verification with	2 months ago
client.h	integrated trusted service access and rsa sign verification with	2 months ago
main.cpp	integrated trusted service access and rsa sign verification with	2 months ago

HTTPS clone URL

https://github.cc

You can clone with HTTPS or Subversion.

Figure 2.3: OFFIS contributions are available on the TRESCCA github account

- **tresccad** is the result of Task 3.3 and implements an off-line migration service to move VMs between heterogeneous platforms. It is documented in D3.3 and D3.4 respectively.
- **Xagent** is a sample application to demonstrate a X11 GUI controlled from inside a VM. It also shows how active migration using tresccad can be achieved, which means triggering the migration process from the application or on users request respectively.
- **trusted_service** is a proof of concept using the TEE implementation of OP-TEE OS to demonstrate a

way how to access crypto services (in this case PKCS signing) of the TEE remotely. A trusted_service application acts as a server and forwards requests to chosen TEE functions.

- **libvirt** contains a patch for libvirt that can be used in combination with tresccad to skip abi validation when restoring guests. This is useful to improve performance of VM migration in tresccad by allowing modification to supplied XML guest configurations.

- **nova** contains patches for OpenStack to use a Raspberry PI as a compute node. It serves as a proof of concept to demonstrate that OpenStack can be installed on a PI and is a requirement to use the guide on the TRESCCA website on how to use [5].

2.1.3 WT

Wellness Telecom disseminated TRESCCA through a collaboration with the FP7 project CUMULUS. As Wellness Telecom is involved in both projects, the dissemination and knowledge transfer between them has been made fluently through our participation. WT has detected during the whole life of the project there are some synergies between both projects as:

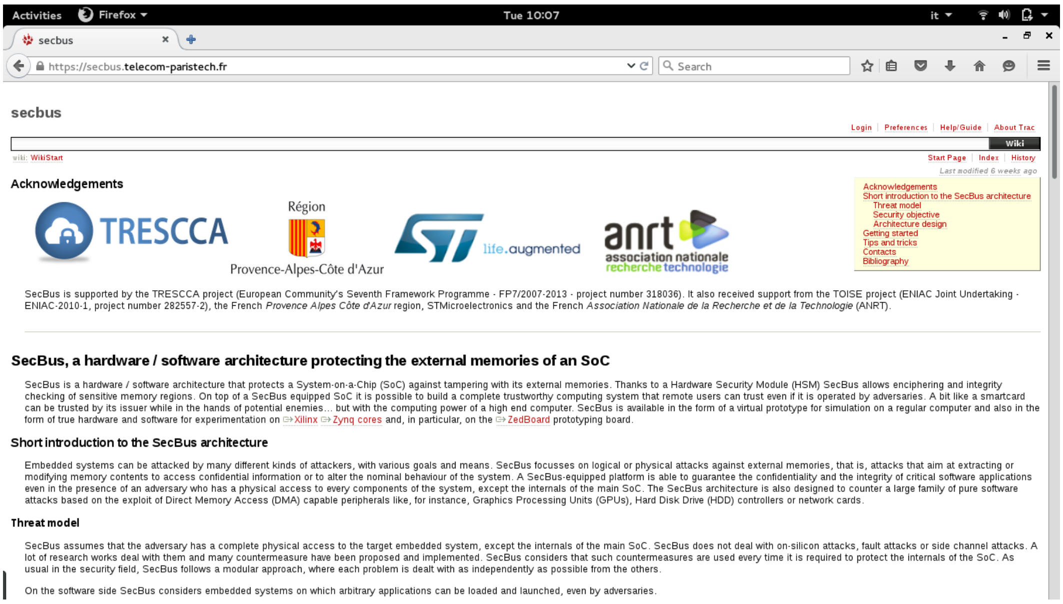
- CUMULUS is developing an integrated framework of models, processes and tools supporting the certification of security properties of infrastructure (IaaS), platform (PaaS) and software application layer (SaaS) services in cloud. CUMULUS could benefit from TRESCCA in the sense of external parties developments that aim to validate the security of new cloud application and services focused on security.

- TRESCCA aims to lay the foundations of a secure and trustworthy cloud platform by ensuring strong logical and physical security on the edge devices, using both hardware security and virtualization techniques while considering the whole cloud architecture. A security certification framework will enable to certificate the security added value that TRESCCA provides to the new devices connected to the cloud.

It would be very interesting if we could have certificated some security properties before and after implementing TRESCCA in the use case we are developing under TRESCCA so both projects would benefit. Nevertheless in TRESCCA we will have different applications and demos ranging from lower level demos to more complete cloud (client/server) based applications and in CUMULUS the framework has not been developed in a generic way so in order to do this testing some extra developing effort and customization need to be done. But in order to have a clear picture of the security properties that CUMULUS could certificate and that TRESCCA is focusing on, we have:

- Authentication.
- Identity and Access Management (IAM).
- Privacy
- Securing Data in Transmission
- User Identity
- Flooding Attacks
- Cloud Integrity and Binding Issues

[5] Raspberry PI as Compute Node in OpenStack: http://www.trescca.eu/index.php/2013-05-23-13-18-38/guides/118-raspberry-pi-as-compute-node-in-openstack.html



Activities
Firefox
Tue 10:07
it
secbus
https://secbus.telecom-paristech.fr
Search
Login | Preferences | Help/Guide | About Trac
Wiki
wiki: WikiStart
Start Page | Index | History
Last modified 6 weeks ago
TRESCCA
Région
Provence-Alpes-Côte d'Azur
life.augmented
anrt
association nationale
recherche technologie

secbus

Acknowledgements

Acknowledgements
Short introduction to the SecBus architecture
Threat model
Security objective
Architecture design
Getting started
Tips and tricks
Contacts
Bibliography

SecBus is supported by the TRESCCA project (European Community's Seventh Framework Programme - FP7/2007-2013 - project number 318036). It also received support from the TOISE project (ENIAC Joint Undertaking - ENIAC-2010-1, project number 282557-2), the French *Provence Alpes Côte d'Azur* region, STMicroelectronics and the French *Association Nationale de la Recherche et de la Technologie* (ANRT).

SecBus, a hardware / software architecture protecting the external memories of an SoC

SecBus is a hardware / software architecture that protects a System-on-a-Chip (SoC) against tampering with its external memories. Thanks to a Hardware Security Module (HSM) SecBus allows enciphering and integrity checking of sensitive memory regions. On top of a SecBus equipped SoC it is possible to build a complete trustworthy computing system that remote users can trust even if it is operated by adversaries. A bit like a smartcard can be trusted by its issuer while in the hands of potential enemies… but with the computing power of a high end computer. SecBus is available in the form of a virtual prototype for simulation on a regular computer and also in the form of true hardware and software for experimentation on Xilinx Zynq cores and, in particular, on the ZedBoard prototyping board.

Short introduction to the SecBus architecture

Embedded systems can be attacked by many different kinds of attackers, with various goals and means. SecBus focusses on logical or physical attacks against external memories, that is, attacks that aim at extracting or modifying memory contents to access confidential information or to alter the nominal behaviour of the system. A SecBus-equipped platform is able to guarantee the confidentiality and the integrity of critical software applications even in the presence of an adversary who has a physical access to every components of the system, except the internals of the main SoC. The SecBus architecture is also designed to counter a large family of pure software attacks based on the exploit of Direct Memory Access (DMA) capable peripherals like, for instance, Graphics Processing Units (GPUs), Hard Disk Drive (HDD) controllers or network cards.

Threat model

SecBus assumes that the adversary has a complete physical access to the target embedded system, except the internals of the main SoC. SecBus does not deal with on-silicon attacks, fault attacks or side channel attacks. A lot of research works deal with them and many countermeasure have been proposed and implemented. SecBus considers that such countermeasures are used every time it is required to protect the internals of the SoC. As usual in the security field, SecBus follows a modular approach, where each problem is dealt with as independently as possible from the others.

On the software side SecBus considers embedded systems on which arbitrary applications can be loaded and launched, even by adversaries.

Figure 2.4: The secbus website

- VM Escape
- Separation between Users
- Securing Data-Storage

2.1.4 TEI

In TRESCCA, TEI and STM have contributed on extending the gem5 framework with time-annotated NoC firewall and gem5 STNoC models. Both the modular gem5 STNoC model (which will be developed further in FP7 DREAMS) and the NoC firewall developed in TRESCCA will be released by TEI to gem5 community http://m5sim.org in June 2016 and accompanied by training videos. The complexity of these modules is estimated around 25K lines of C++ code. Although RTL code of the NoC firewall developed by TEI in TRESCCA is released only under an NDA, a binary version that runs linux and includes system drivers and testbenches is available for experimentation upon request.

2.1.5 TP

The virtual prototype of the HSM-Mem (SystemC model based on SoCLib), the VHDL model (with its simulation and synthesis environments) of the HSM-Mem, and the software driver for MutekH are released under the CeCILL free and open-source license [6]. They can be downloaded from TP public website: https://secbus.telecom-paristech.fr.

[6] Text of the CeCILL license: http://www.cecill.info/licences/Licence_CeCILL_V2.1-en.txt

3 CONCLUSION

The TRESCCA consortium has always considered open source, standardization and other project dissemination activities of pivotal importance to successfully achieve a wide adoption of the developed technology. This document is the report of the efforts spent by the TRESCCA partners in this direction.

During the three years of the project, many of the TRESCCA technologies has been made publicly available (e.g., KVM-TCG hybrid migration, HSM-mem VHDL model, tresscad, etc.) and other will be released in future (i.e., NoC Firewall). Moreover, standardization initiatives (e.g., GlobalPlatform Trusted Execution Environment) have been continuously monitored and in some cases extended, in the context of the project, by the TRESCCA partners. Finally, the dissemination and the mutual cooperation with other European Commission funded projects has not been neglected, as the TRESCCA consortium believes that sharing experiences and results is beneficial for both projects.