D3.2
Basic NED implementation

Project number	611458
Project acronym	SECURED
Project title	SECURity at the network EDge
Project duration	36 months (1/10/2013–30/9/2016)
Programme	FP7 (Collaborative Project)

Deliverable type	**P** - Prototype
Deliverable number	D3.2
Version (date)	v1.0 (4/5/2015)
Work package(s)	WP3
Due date	31/3/2015 – M18

Responsible organisation	UPC
Editor	Rene Serral Gracia
Dissemination level	**PU** - Public

Abstract	This report describes the implementation and setup instructions for the first prototype of the NED.
Keywords	NED, prototype, documentation, interfaces

Editor
Rene Serral Gracia (UPC)

Reviewers
Antonio Lioy (POLITO)

Contributors
Francesco Ciaccia (BSC)
Roberto Bonafaglia (POLITO)
Tao Su (POLITO)
Roberto Sassu (POLITO)
Adrian L. Shaw (HPLB)
Fulvio Risso (POLITO)
Jarkko Kuusijarvi (VTT)

Acknowledgement
This work was partially supported by the European Commission (EC) through the FP7-ICT programme under project SECURED (grant agreement no. 611458).

Change Log

Version	Date	Note	Author
v1.0	13.04.2015	Version ready for quality control	A.Shaw (HPLB)
v1.1	04.05.2015	Quality control	A.Lioy (POLITO)

Executive Summary

This document illustrates the Network Edge Device (NED) prototype delivered at M18, which includes the backend APIs and the compartmentalization and trust architectures defined in task T3.1. The NED results in a purely software component, so that it can be used autonomously as a virtual-NED (to be used in cases where SECURED networking hardware is not available) or coupled with an OpenFlow-capable network switch to implement the split-NED model. This will permit a fast implementation of the basic concepts in order to offer a foundation platform for the development of the other SECURED components. This task provides also an initial basic version of the Personal Security Controller Manager (PSCM) and the Trusted Virtual Domain Manager (TVDM) that supports the Personal Security Controller (PSC) activation and control, and simple service graph instantiation.

Contents


SECURED

1 Introduction

This prototype demonstrates the basic building blocks of the SECURED NED. Primary features of this prototype include (Fig. 1):

- remote attestation of NED using a remote verifier and attestation-enabled strongSwan client;
- secure and trusted dataplane using strongSwan IPsec and Integrity Measurement Architecture (IMA);
- web-based user authentication (username and password);
- user service graph (SG) parsing and dataplane setup;
- automated instantiation of a user's Trusted Virtual Domain (TVD)
  - Personal Security Controller (PSC) instantiation;
  - instantiation of Personal Security Applications (PSAs) in the service graph;
  - customised configurations for the IPtables PSA.

External SECURED services (e.g. the PSA repository, User Profile repository and the Security Policy Manager) are not included as part of this prototype. These are to be developed as part of the activities of Work Packages 4 and 5. In this prototype, PSA images, service graphs and policies are all cached on the NED.

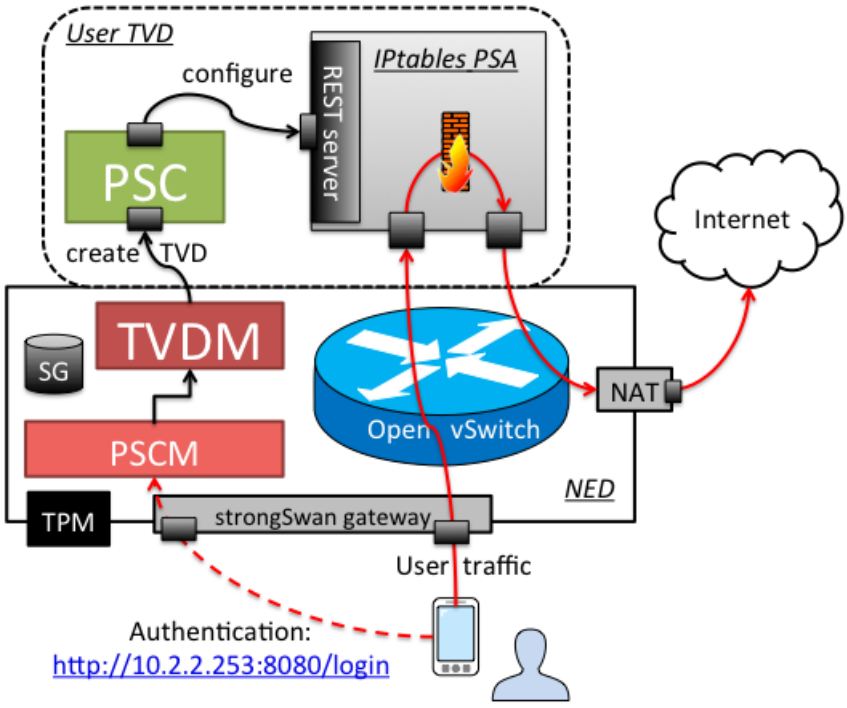

Figure 1: Prototype instantiation of the NED architecture, hosting a simple TVD

The key NED components are described below (for full details, please refer to the SECURED architecture document D2.3.1):

Personal Security Controller Manager (PSCM) is a web-based frontend to the NED and is responsible for authenticating the user. Upon successful user authentication, the PSCM sends a request to the Trusted Virtual Domain Manager (TVDM) to instantiate the user's security. It is accessible from an ordinary client web browser at `http://10.2.2.253:8080/login`. In the future, the user experience may be improved by using a captive portal.

Trusted Virtual Domain Manager (TVDM) is a privileged component within the NED which is able to create and orchestrate trusted virtual domains (TVD). A TVD is a logical security domain given to an individual tenant, providing traffic isolation and application isolation. The TVDM enforces the isolation between TVDs by controlling the NED's virtual switch. OpenvSwitch was used as the software switch in this prototype and is the keystone for providing multi-tenancy and isolation. Whilst TVD is a term which is agnostic to specific technologies, in this version of the prototype a user TVD consists of a set of small virtual machines. There are also parallel efforts within the SECURED project to use Docker (Linux) containers within a TVD, which provide less computational and space overheads.

Personal Security Controller (PSC) is the first entity created in a user's TVD. It is responsible for parsing the service graph **(SG)** and informing the TVD about which computational resources are required. Once the PSA environments are instantiated, the PSC controls the PSAs at runtime over a private control network within the TVD.

Personal Security Application (PSA) is an environment consisting of all the dependencies required to running a personalised security application. It includes a REST server which allows it to be configured and controlled. In this version of the prototype, a PSA is a virtual machine image (raw IMG file) which has two virtual network interface cards (vNICs) for the dataplane (ingress/egress) and a third vNIC for the control network (where the HTTP REST server listens on).

2 Setup guide and package dependencies

CentOS is the fully supported GNU/Linux distribution for SECURED[1], although some parts can work on other distributions with the same packages and minor tweaking.

Install the packages for CentOS:

```
# yum install python-pip curl unzip virt-manager wget openvswitch
    python-futures rsync qemu-kvm.x86_64 virt-manager gcc gmp-devel
    unzip trousers trousers-devel java-1.7.0-openjdk java-1.7.0-
    openjdk-devel
```

Then use pip to get the latest Python modules:

```
  # pip install falcon requests gunicorn trollius lxml
```

[1]This platform has been selected because it includes an up-to-date database with the signatures of all the supported software, which is a requisite for the remote attestation.

Finally, you must decide how the users will connect to the NED. You have two options: IPsec only or without IPsec. Note that the prototype does not allow mixed access connections (this is an improvement planned for T3.3).

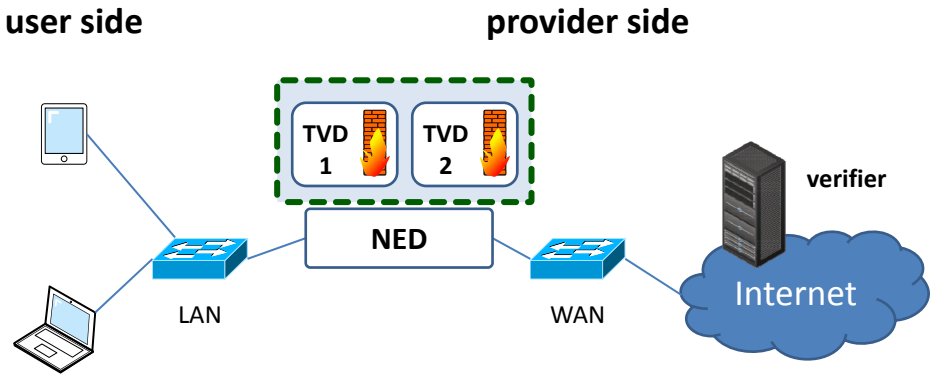

Figure 2: Supported and tested topology for the basic NED.

3 IPsec-only connections (remote NED scenario)

If you want all users to connect to the NED over IPsec, then you can just execute the following script on the NED (as root user):

```
# ./setup.sh your_internet_facing_interface
```

Where <*your_internet_facing_interface*> could have been `eth0`, for example.

4 IPsec-less connections (home/IoT scenarios)

Alternatively, if the user is in a private closed (and trusted) environment (home scenario, IoT, etc) and the NED is the immediate gateway, then the user will need a DHCP service to provide an IP on the dataplane. In the current version of the prototype, this is done by the NED itself, but in future it could support an external DHCP server. To allow access to the NED without using IPsec you will need to specify the user-facing Ethernet port as well:

```
# ./setup.sh your_internet_facing_interface user_facing_port
```

Example: `sudo ./setup.sh eth0 eth1`

The user will then be able to access the PSCM from outside the NED at: `http://10.2.2.253:8080/login`

There may be a captive portal in future to redirect to this page.

5 User profiles

Once at the login page (`http://10.2.2.253:8080/login`) you can login with one of the following test profiles:

- **user1:** when logging in will start a PSA which should block all traffic to www.polito.it
- **user2:** when logging in will start a PSA which should block all traffic to www.upc.edu

Both accounts have the following password: **secuser**

6 Tests

If a problem occurs, unit tests can be applied to all the major components. For functional testing the BATS (Bash Automated Testing System) framework is needed. To install it:

```
$ git clone https://github.com/sstephenson/bats.git
$ cd bats
# ./install.sh /usr/local
$ cd ..
```

Then to perform the tests (must be root or else it will fail):

```
# bats ./test.bats
```

You should then get an output for tests against each component, such as:

```
- Test PSCM reachability
- Test TVDM reachability
- Test PSCM authentication
- Test PSC's VM reachability
```

If you wish to test the browser interface from inside the NED you can use the following command (replace firefox with your favourite browser, e.g. lynx):

```
ip netns exec pscmNs firefox 10.2.2.253:8080/login
```

7 IPsec with IMA, remote verifier and the NED RA agent

This only works on CentOS. Make sure the TPM is enabled in the BIOS. Clone and set up the verifier on a machine in the same network. The project is available ta:

https://gitlab.secured-fp7.eu/secured/verifier

The code is used to create a Third-Party Verifier with OpenAttestation v1.7 and ratools developed by TorSec Group in Politecnico di Torino and it requires GNU/Linux CentOS 7.

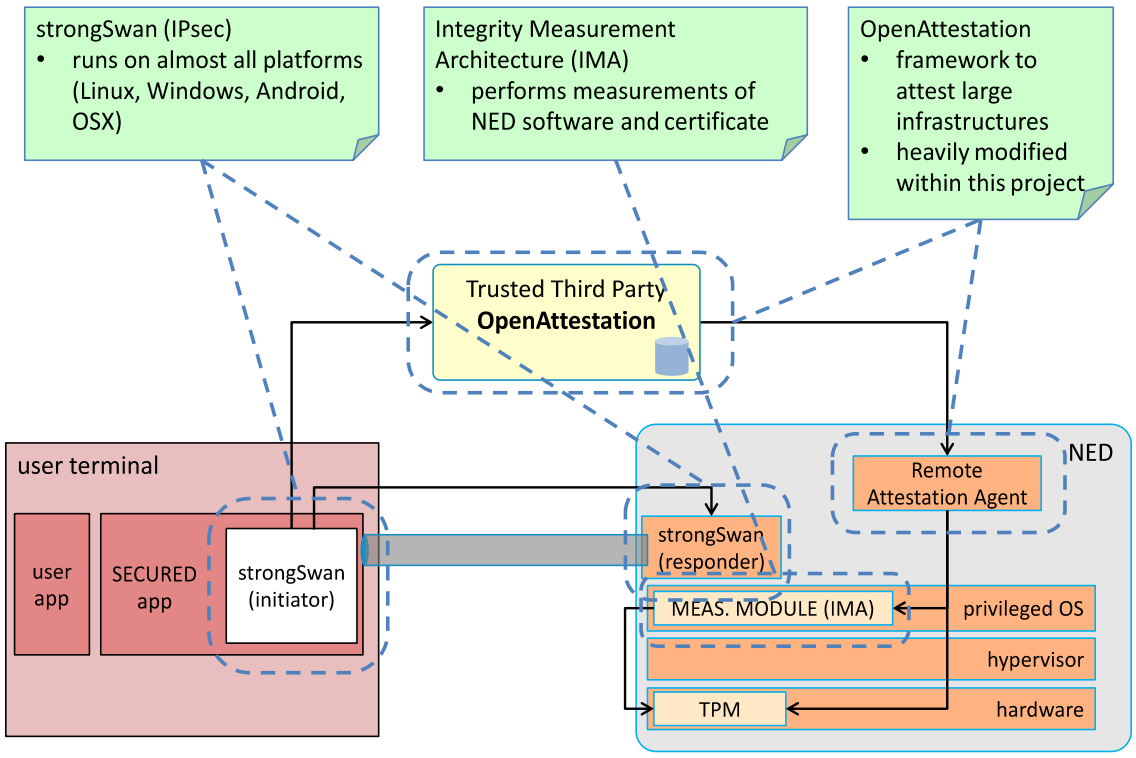

Figure 3: Overview of trusted channel support (full details in D3.2.1).

7.1 Verifier setup

Install the following packages:

```
yum install ant, trousers, trousers-devel, php-soap, mariadb, mariadb-
    server, python-networkx, python-suds, python-mathplotlib, graphviz
    -devel
```

In addition:

```
yum install patch, java-1.7.0-openjdk, java-1.7.0-openjdk-devel, zip,
    unzip, gcc, gcc-c++, rpm-build, python, python-devel, python-pip
```

Steps to configure the verifier:

- map the host names with the hardcoded IP addresses;
- download OpenAttestation branch v1.7 from the official repository `https://github.com/OpenAttestation/OpenAttestation.git`
- apply the patch `0001-HisAppraiser-Fixed-reading-of-host-name-from-the-rep.patch` in the OpenAttestation/Source directory, then run

```
download_jar_packages.sh
distribute_jar_packages.sh
```

- move to the Installer folder and run `rpm.sh`. Please note that this command has to be invoked exactly from the above directory, otherwise it does not work. If `bash` does not work, please use `sh` instead.
- install the package in `/root/rpmbuild/RPMs/x86_64/`
- perform the next step only if `mariadb` service is running. In fact, since mysql is not available in CentOS7, you need to change to mariadb in the .spec file.
- configure OpenAttestation with the configure script with this format

```
bash configure_oat.sh $selfname $attestorname $attestorIP
    $PCR0value $OSdistname
```

  You need to modify in `configure_oat.sh` the path to the ra_verifier. For example:

```
bash configure_oat.sh verifier ned xxx.xxx.xxx.xxx 7
    D94A15BE0295A3743FC259B07202FF42550B369 CentOS7
```

- open ports to receive and send integrity reports; default ports are 80 and 8443;
- change OAT.property to store integrity reports in files, receive delta reports and discard identical integrity reports; uncomment the following:

```
IR_DIR, IR_DIGEST_METHOD, SCALABILITY, DISCARD_IDENTICAL_IR
```

7.2 NED setup

Now perform the following steps:

- go back to the NED, enter the strongSwan directory and run the following as root

```
# ./ipsec.sh remote_verifier_ip_address ned_ip_address ned_hostname
```

- reboot;
- start the commands in CommandTool to poll the integrity reports and verify it as defined in the configure.sh script;

```
$ oat_pollhosts -h verifier '{"hosts":["ned"],"analysisType":"
    load-time+check-cert,l_req=l4_ima_all_ok|==,cert_digest=095
    b7792c076d65a9c45f4f484d06cd1fa29a9ba"}'
```

7.3 Building the remote attestation client for end users

Install dependencies:

```
yum install gcc rmp-build yum-utils
```

Configuration steps:

- map hardcoded IP addresses with the verifer and NED;
- download the source code of strongswan from the EPEL repository and patch it

```
yum install http://mirrors.nic.cz/epel/7/x86_64/e/epel-release
    -7-5.noarch.rpm
yumdownloader --source strongswan
rpm -Ui strongswan-5.2.0-7.el7.src.rpm
cd rpmbuild/SPECS/
nano strongswan.spec (add oat-attest, soup plugin and ra.patch)
rpm -Uivh strongswan-5.2.0-7.el7.src.rpm
```

- build the package and install it; spec file is uploaded
- change the IPsec configuration (`ipsec.conf` and `ipsec.secrets`);
- exchange the certificates with the NED;
- configure the `oat_attest.conf` file to define the verifier and the appraiser;
- configure the file `curl.conf` to disable the load option.

8 Credits

Lead developers:

- Francesco Ciaccia (BSC)
- Roberto Bonafiglia (POLITO)

Remote Attestation and IPsec developers:

- Roberto Sassu (POLITO)
- Tao Su (POLITO)

Documentation, testing and validation:

- Adrian L. Shaw (HPLB)
- Jarkko Kuusijarvi (VTT)