SEAPATH and cybersecurity

SEAPATH follows the applicable cybersecurity guidelines defined by the ANSSI in the ANSSI-BP-028 document . Several mechanisms have been taken into account in order to guarantee system's security:

- System hardening
- Disk encryption
- Secrets storage and protection
- Process isolation
- Privileges management policy
- Connection encryption
- User authentication process

Yocto Project

The Yocto version of SEAPATH follows an earlier version of the ANSSI-BP-028 document: ANSSI-BP-028 v1.2.

The compatibility matrix can be found after the build in build/security/traceability-matrix_seapath-guest-efi-image_ANSSINT28.adoc.

TODO: Write an explanation on the current state of each requirements on Yocto. This should be automatically done when merging the cukinia test of Yocto and Debian.

Debian

The Debian version of SEAPATH follows the version 2 of the document : ANSSI-BP-028 v2

A compliance matrix listing all the tests done on SEAPATH and their relation to the requirements is available at the end of each test report on the CI. You can find weekly test reports here: https://github.com/seapath/ci/tree/reports-PRmain/docs/reports/PR-main

Below is a detailed list of all requirements, their current state on SEAPATH and a small explanation on the work done or to be done.

- Done: SEAPATH complies with this requirement. Tests are run with cukinia to ensure that future development don't break this compliance. (Some requirements are done, but no tests exist for them. When it is so, it is explicitly written in the table below.)
- Not Done: SEAPATH doesn't comply with this requirement. A small description of the work to do is given.
- Not applicable: This requirement has no sense to be applied on SEAPATH.
- User applicable: This requirement cannot be fulfilled by SEAPATH and must be ensured by the end user/SEAPATH integrator.
- Partially done: This requirement is not done in SEAPATH. However, some specific parts of the requirement are done, and tests exists for it.

	Subject	Level	Explanations	State
R1	Choosing and configuring the hardware	MIE	The hardware chosen to run SEAPATH must comply with https://cyber.gouv.fr/publications/recommandations-de-configuration-materielle-de-postes-clients-et-serveurs-x86 Note that all modern x86 machines are already compatible. The ANSSI has not released a similar Document for ARM machines.	User applicable
R2	Configuring the BIOS/UEFI	MI	The BIOS must be configured according to the document https://cyber.gouv.fr/publications/recommandations-de-configuration-materielle-de-postes-clients-et-serveurs-x86	User applicable
R3	Activating the UEFI secure boot	MI	SEAPATH is compatible with Secure Boot and support preload keys. No test exists currently	User applicable
R4	Replacing of preloaded keys	MIEH	We must replace the preload keys with ours. It implies : - Generating a set of UEFI keys and storing them in a secure location, ideally a Hardware Security Module (HSM) - Signing the kernel, Grub, all kernel modules and certain firmware files - Integrate these newly signed files into build_debian_iso - Sign all future kernel, Grub and kernel module updates - Own APT repositories to store these new signed update files	Not Done
R5	Configuring a password on the bootloader	MI	Grub password can be configured in ansible inventory.	Done
R6	Protecting the kernel command line parameters and the initramfs	MIEH	Currently, neither the kernel nor the initramfs are protected by secure boot.	Not Done
R7	Activating the IOMMU	MIE	IOMMU must be activated in force mode. Non-regression tests must be performed to ensure that this is compatible with SEAPATH features	Not Done
R8	Configuring the memory options	MI	The options are present on SEAPATH. The related test checks that the CPU has no known vulnerabilities	Done
R9	Configuring the kernel options	MI	The kernel options are present	Done

R 10	Disabling kernel modules loading	M IE	The goal is to deactivate module loading once all desired modules are loaded. The de-activation is simple to do, but we must think of a policy to detect that all desired modules are loaded.	Not Done
R 11	Configuration option of the Yama LSM	MI	The kernel parameter security=yama is present. The sysctl is configured to 2	Done
R 12	IPv4 configuration options	MI	IPV4 must comply to a certain list of sysctl configuration. Some sysctl are natively enabled, but not all are tested correctly. The rest of the sysctl must be activated by taking care of not breaking a SEAPATH feature. A reason must be explicitely given for the sysctl that cannot be activated.	Partially done
R 13	Disabling IPv6	MI	IPV6 is not used on SEAPATH. It is disabled in the kernel parameters.	Done
R 14	File system configuration options	MI	The recommended options are present on SEAPATH.	Done
R 15	Compile options for memory management	M IEH	We have to recompile our own kernel. This implies: - Following upstream corrections - Managing our own UEFI keys - Recompile and sign all Linux modules - Manage our own APT repositories This is a huge work to do, considering R4 already done.	Not Done
R 16	Compile options for kernel data structure	M IEH	This recommendation implies recompiling the kernel. The work is the same as R15.	Not Done
R 17	Compile options for the memory allocator	M IEH	This recommendation implies recompiling the kernel. The work is the same as R15.	Not Done
R 18	Compile options for the management of kernel modules	M IEH	This recommendation implies recompiling the kernel. The work is the same as R15.	Not Done
R 19	Compile options for abnormal situations	M IEH	This recommendation implies recompiling the kernel. The work is the same as R15.	Not Done
R 20	Compile options for kernel security functions	M IEH	The Debian kernel used by SEAPATH is already compiled with these options. TODO : Add a test for this	Done
R 21	Compile options for the compiler plugins	M IEH	This recommendation implies recompiling the kernel. The work is the same as R15.	Not Done
R 22	Compile options of the IP stack	M IEH	This recommendation implies recompiling the kernel. The work is the same as R15.	Not Done
R 23	Compile options for various kernel behaviors	M IEH	This recommendation implies recompiling the kernel. The work is the same as R15.	Not Done
R 24	Compile options for 32 bit architectures	M IEH	This recommendation targets 32 bits x86 machines. Currently, SEAPATH is not tested on such hardware.	Not Done
R 25	Compile options for x86_64 bit architectures	M IEH	This recommendation implies recompiling the kernel. The work is the same as R15.	Not Done
R 26	Compile options for ARM architectures	M IEH	This recommendation targets ARM based processor. Currently, SEAPATH is not tested on such hardware.	Not Done
R 27	Compile options for ARM 64 bit architectures	M IEH	This recommendation targets ARM based processor. Currently, SEAPATH is not tested on such hardware.	Not Done
R 28	Typical partitioning	MI	Currently, only /boot and / are separated.	Not Done
R 29	Access restrictions on /boot	M IE	/boot is restricted to root, but is always mounted. To fulfill this requirement, we must develop a mechanism that mount /boot only when needed and unmount it after that. Note that cukinia launch tests on /boot and will need it to be mounted.	Not Done
R 30	Removing the unused user accounts	M	There are no unused accounts on SEAPATH	Done
R 31	User password strength	M	The passwords used on SEAPATH must follow https://www.ssi.gouv.fr/mots-de-passe/ The only password used on SEAPATH is define during the build of the Debian ISO. Additionnal passwords added by PAM software must also follow this recommendation.	User applica ble
R 32	Configuring a timeout on local user sessions	MI	The bash timeout is set to 300s.	Done
R 33	Ensuring the imputability of administration actions	MI	Only sudo commands are logged.	Not Done
R 34	Disabling the service accounts	MI	No additionnal accounts can be opened by a service on SEAPATH.	Done
R 35	Uniqueness and exclusivity of service accounts	MI	Not all services have a dedicated account.	Not Done

R 36	Changing the default value of UMASK	M IE	UMASK is set to the desired value.			Done
R 37	Using Mandatory Access Control features	M IE	SEAPATH uses Apparmor, the MAC solution of Debian.			Done
R 38	Creating a group dedicated to the use of sudo	M IE	The group « privileged » is used for sudo usage. If a PAM authentification is implemented by the end user, privileged users must also use this group.			Done
R 39	Sudo configuration guidelines	MI	All desired options are implemented and tested.			Done
R 40	Using unprivileged users as target for sudo commands	MI	No command targets root.			Done
R 41	Limiting the number of commands requiring the use of the EXEC directive	M IE	Commands allowed to run with sudo should not used the EXEC directive. The are two exceptions currently in SEAPATH: • Ansible user needs to spawn shells with root privilege. • Admin user can run all commands as root There is currently no specific policy to handle the ansible user after the initial configuration, but the end user could think about removing or deactivating the user when it is not needed. The admin user should probably be split in many users regarding what they are supposed to do (observation only, handle VM, handle updates ...) TODO			Done
R 42	Banishing the negations in sudo policie	MI	No negation is present in the sudoer files			Done
R 43	Defining the arguments in sudo specifications	MI	When possible, all commands allowed to run with sudo must define specific arguments. There are two exceptions currently in SEAPATH: ▪ Ansible user needs to access a shell with root privilege. ▪ Admin user can run all commands as root The remarks are the same for R41. TODO			Done
R 44	Editing files securely with sudo	MI	No text editor must be launched with sudo privileges. To modify the sudoers rules, the visudo command is installed on SEAPATH. Note that sudo rules should not be changed after the initial configuration of SEAPATH			User applica ble
R 45	Activating AppArmor security profiles	M IE	All AppArmor profiles are present, but no test exists for it.			Done
R 46	Activating SELinux with the targeted policy	M IEH	Debian uses AppArmor instead of SELinux			Not Applica ble
R 47	Containing the unprivileged interactive users	M IEH	Debian uses AppArmor instead of SELinux			Not Applica ble
R 48	Setting up the SELinux variables	M IEH	Debian uses AppArmor instead of SELinux			Not Applica ble
R 49	Uninstalling SELinux Policy Debugging Tools	M IEH	Debian uses AppArmor instead of SELinux			Not Applica ble
R 50	Limiting the rights to access sensitive files and directories	MI	This principle is followed natively by Debian. TODO : Write a list of sensitive directories and test that they have the correct access rights.			Done
R 51	Changing the secrets and access rights as soon as possible	M IE	SEAPATH is meant to be entirely functional once the installation and configuration is completed. TODO : Is it possible to write a test for that ?			Done
R 52	Securing access for named sockets and pipes	MI	SEAPATH comply with this recommendation, but the test is difficult to write. TODO : write a test for that			Done
R 53	Avoiding files or directories without a known user or group	M	All files and directory have a known user and group			Done
R 54	Setting the sticky bit on the writable directories	M	The sticky bit is set for all writable directories			Done
R 55	Dedicating temporary directories to users	MI	All users and services have a dedicated temporary directory.			Done
R 56	Avoiding using executables with setuid and setgid rights	M	No executables added by the SEAPATH project have the setuid or setgid rights. This recommendation is not applicable to Debian native executables.			Done
R 57	Avoiding using executables with setuid root and setgid root rights	M IE	Some executables still have root setuid and setgid.			Not Done
R 58	Installing only strictly necessary packages	M	A list of necessary packages is described in the testing process. A test verifies that no additionnal packages are installed.			Done
R 59	Using only official package repositories	M	Only Debian repository are used by default. The end user can add its own mirror during the Ansible configuration.			Done

R60	Using hardened package repositories	M IE	Debian don't use hardened packages repositories	Not Applicable
R61	Updating regularly the system	M	SEAPATH provides an update system. On Debian, apt updates are used. Refer to this page for more information https://wiki.lfenergy.org/display/SEAP/IT+tooling	User applicable
R62	Disabling the non-necessary services	M	A list of necessary services is described in the testing process. A test verify that no additionnal services are started.	Done
R63	Disabling non-essential features of services	MI	We must take the list of services done in R62 and limit the functionnalities of all services to the minimum required.	Not Done
R64	Configuring the privileges of the services	M IE	A complete list of the services and their privileges must be established in order to restrict the services that can be and justify why others cannot.	Partially done
R65	Partitioning the services	M IE	Many services are already hardened, but not all. A complete list of the services and their hardening must be established in order to harden the services that can be and justify why others cannot.	Partially done
R66	Hardening the partitioning components	M IEH	This means hardening docker, KVM/QEMU and SystemD. QEMU is already hardened by SystemD, but the recommendation does not give any limits or advice on its application. Docker is not hardened.	Not Done
R67	Secure remote authentication with PAM	MI	The Kerberos protocol can be installed on SEAPATH during the build of the Debian ISO. The choice and configuration of the software used to handle remote login must be made by the end user.	User applicable
R68	Protecting the stored passwords	M	The password storage must follow https://www.ssi.gouv.fr/mots-de-passe/. Tests verify that the initial and generated passwords complies.	Done
R69	Securing access to remote user databases	MI	Similar to R69, this part must be configured by the end user after selecting the remote login software.	User applicable
R70	Separating the system accounts and directory administrator	MI	The selection of the users and their rights is highly dependent of the final use case of SEAPATH. Additionnal user can be created in ansible debian role. Their rights must be configured in the sudoers file.	User applicable
R71	Implementing a logging system	M IE	SEAPATH uses journald for local logs and syslog-ng for remote logs. These systems doesn't fully comply with the recommendation.	Partially done
R72	Implementing dedicated service activity journals	M IE	SEAPATH uses systemd that complies with this requirement.	Done
R73	Logging the system activity with auditd	M IE	Auditd is installed and configured.	Done
R74	Hardening the local messaging service	MI	SEAPATH don't have any messaging service	Not Applicable
R75	Configuring aliases for service accounts	MI	SEAPATH don't have any messaging service	Not Applicable
R76	Sealing and checking files integrity	M IEH	We must install intrusion monitoring tools	Not Done
R77	Protecting the sealing database	M IEH	The sealing database is generally protected by intrusion monitoring tools.	Not Done
R78	Partitioning the network services	M IE	It is the goal of SEAPATH to isolate services on virtual machines (or containers). However, some services remains not isolated and it will be very complicated to isolate some (eg : snmp, ceph, ssh ...)	Not Done
R79	Hardening and monitoring the exposed services	MI	Network services are already hardened by systemD. However, this recommendation doesn't precise a limit to the hardening. A list of the hardening measure of systemd must be done and made available to SEAPATH integrators	Partially done
R80	Minimizing the attack surface of network services	M	All network sockets listen on a dedicated interface.	Done