Huawei user authentication solution “RTOS”

SECURITY TARGET

Version 1.3

February 2021
Document Reference: HRTOS_ST_1.3

DOCUMENT HYSTORY		
Revision	Date	Changes description
1.0	20-Jul-20	Validation of 1st version
1.1	20-Oct-20	Version Number Update
1.2	29-Oct-20	Document name change
1.3	02-Feb-21	Typo change within table on page 11

Table of contents

Abbreviations

Abbreviation	Description
AH	Authentication Header
DAC	Discretionary Access Control
HMAC	Hash-based Message Authentication Code
RNG	Random number generator.
RTOS	Real-Time Operating System
SoC	System on Chip
SSH	Secure Shell
UID	User identifier, is a number assigned by Linux to each user on the system
GID	Group Identifier

1 IDENTIFICATION OF THE EVALUATED PRODUCT

1.1 Reference and version of the evaluated product

Developer	Huawei
Product name	RTOS
Evaluated version	207.6
Technical domain	Identification, authentification et contrôle d'accès + Communication sécurisée

1.2 Procedure to identify the evaluated product

There is a file "/etc/RTOS-Release", with 0644 permission mode, which contains Huawei-RTOS version information.

By running 'cat /etc/RTOS-Release', users can get information about RTOS product.

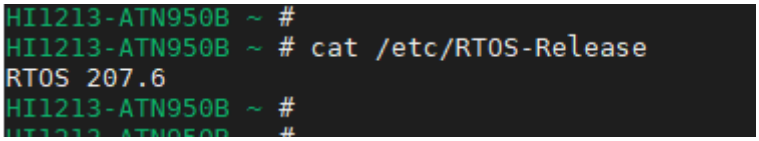

Figure 1: Checking the product version

2 PRODUCT DESCRIPTION

2.1 General Product description

The product specified by the present security target is Huawei RTOS, a highly-configurable Linux-based operating system for embedded devices, which has been developed to provide a good level of security as required in commercial environments.

RTOS is a general purpose; multi-user, multi-tasking Linux based operating system. It provides a platform for a variety of applications.

RTOS evaluation covers a potentially distributed network of systems running the evaluated version and its configurations as well as other peer systems operating within the same management domain.

The Security Functions consist in functions of RTOS that run in kernel mode plus some trusted processes running in user mode. These are the functions that enforce the security policy as defined in this Security Target. Tools and commands executed in user mode that are used by an administrative user need also to be trusted to manage the system in a secure way, but they are not considered to be part of the security functions.

The hardware, BIOS firmware and potentially other firmware layers between the hardware and the RTOS are considered to be part of the RTOS environment.

The RTOS includes standard networking applications, such as sshd, which allows accessing the RTOS via cryptographically protected communication channel.

System administration tools include the standard command line tools. The graphical user interface for system administration or any other operation is not included in the evaluated configuration.

The RTOS environment also includes applications that are not evaluated, but are used as unprivileged tools to access public system services. For example, a network server using a port above 1024 may be used as a normal application running without root privileges on top of the RTOS. Additional documentation is available and provides guidance on the way to set up such applications on the RTOS in a secure way.

2.2 Functionalities of the Product

RTOS provides the following key security features:

- Identification and Authentication: The RTOS includes several ways to identify and authenticate the users (via the local console using username and password or via the SSH using password and public-key based authentication. The RTOS also offers a password quality enforcement mechanism as well as it is able to handle failed authentication attempts, accordingly to recommendations of [LINUX]. This feature includes also the ability for the RTOS to end user sessions after a period of inactivity
- Trusted Channel: Using the cryptographic communication protocols above mentioned (SSH) the RTOS is able to establish secure and trusted communication channel to and from other IT entities.
- Segregating users and configuration data: The RTOS offers to the users and/or authorized administrators the ability to modify the configuration of the RTOS and a Discretionary Access Control (DAC) which allow owner of named objects controlling the access permissions to these objects and enforcing configuration data that is accessed only by granted users.

2.3 Product usage

The RTOS is a Linux-based multi-user multi-tasking operating system. It may provide services to several users, local or remote, at the same time. After successful login, the users get access to a general computing environment, allowing launching user applications, issuing user commands at shell level, creating and accessing files. It provides adequate mechanisms to separate the users and protect their data. Privileged commands are restricted to only administrative users.
The RTOS is also intended to operate in a networked environment with other instantiations of the RTOS as well as other well-behaved peer systems.

It is assumed that responsibility for the safeguarding of the user data protected by the RTOS can be delegated to human users of the RTOS if such users are allowed to log on and spawn processes on their behalf. All user data is under the control of the RTOS. The user data is stored in named objects, and the RTOS can associate a description of the access rights to that object with each named object.

The RTOS enforces controls such that access to data objects can only take place in accordance with the access restrictions placed on that object by its owner, and by administrative users. Ownership of named objects may be transferred under the control of the access control policies implemented by the RTOS.

The RTOS enforces discretionary access control policy, in which, access rights (e.g. read, write, execute) can be assigned to data objects with respect to subjects identified with their UID, GID and supplementary GIDs. Once a subject is granted access to an object, the content of that object may be used freely by the subject to influence other objects accessible to the same subject.

2.4 Technical environment of the product

All security functions claimed in this security target that apply to physical devices compatible with the ARM64 V8 instruction set are not part of the evaluation scope.

The following physical hardware platforms, corresponding firmware, and components are supported by the RTOS:

- Hi1213 Soc based hardware device board
- Hi1610 Soc based hardware device board
- SD5573 Soc based hardware device board
- Hi1383 Soc based hardware device board
- SD8081 Soc based hardware device board

The evaluation scope is the RTOS for the Hi1213 Soc based hardware device board only.

2.5 Evaluation platform

The RTOS is tested on a Hi1213 Soc based hardware device board. This board contains physical peripheral devices (flash storage, network interface cards, serial interfaces) which can be used with the RTOS without affecting its security functions.

The figure below shows the process to customize and build the RTOS:

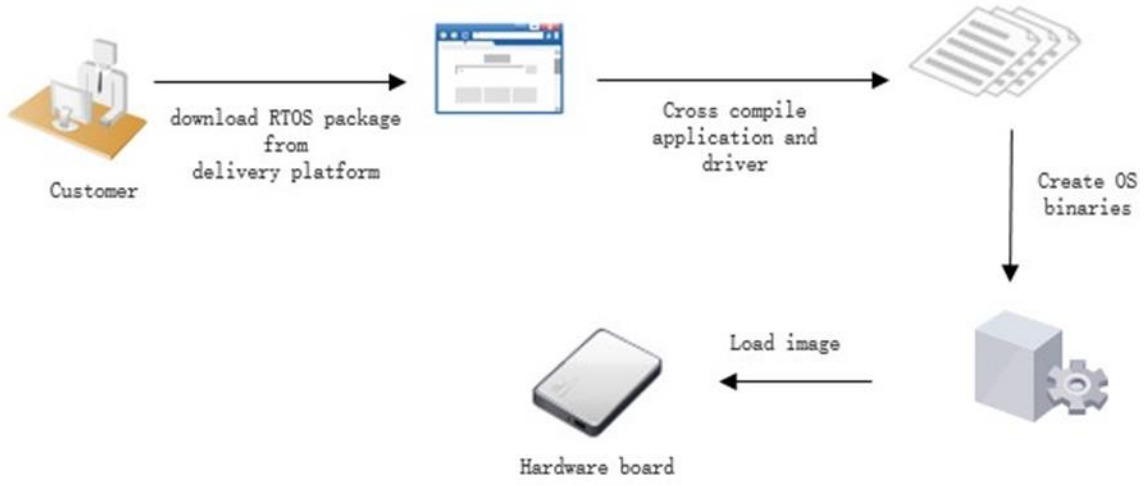

Figure 2: Customizing and building process

The cross compilation environment is a host machine installed with SUSE Linux enterprise server 12 sp5 x86_64.

3 OPERATIONAL ENVIRONMENT OF THE PRODUCT

3.1 Product users

Through its external interfaces, the RTOS support two users' profiles, which are:

- Regular users (shortly users)
- Administrators

Note: Users with "su" permissions are considered as administrators for the commands they are allowed to run with root privileges and only for these commands.

3.2 Product exploitation environment and assumptions

Complementary to the general product description in the section 2.1, the following assumptions apply to the RTOS environment.

3.2.1 Physical aspects

A.PHYSICAL	It is assumed that the IT environment provides the RTOS with appropriate physical security, commensurate with the value of the assets protected by the RTOS.

3.2.2 Personnel aspects

A.MANAGE	The RTOS security functionality is managed by one or more competent individuals. The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the guidance documentation.
A.AUTHUSER	Authorized users possess the necessary authorization to access at least some of the information managed by the RTOS and are expected to act in a cooperating manner in a benign environment.
A.TRAINEDUSER	Users are sufficiently trained and trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their user data.

3.2.3 Procedural aspects

A.PEER.MGT	All remote trusted IT systems and users are assumed to be under the same management control and security policy constraints.
A.PEER.FUNC	All remote trusted IT systems are assumed to correctly implement their functionalities.

3.2.4 Connectivity aspects

A.CONNECT	The connections between a remote trusted IT systems using the serial interface of the hardware platform are physically or logically protected within the RTOS environment to ensure the integrity and confidentiality of the data transmitted and to ensure the authenticity of the communication end points.

3.3 Protection offered by the product's environment

The assumptions of the previous section allow ensuring that:

- No physical attack can be done on the devices hosting the RTOS.
- No spying or intrusion can been done on a legitimate console connection through the serial port.
- Administrator users are not attackers for the RTOS assets.

4 ASSETS

4.1 User assets

None users' assets are included in the default configuration of the product.

Only the home directory of the default regular user "rtos" is configured with the correct owner, group and privileges, as shown in the table of the next section.

4.2 Product assets

The product assets are the functions implemented by the software product and the related configuration.

The security needs for the function are integrity and availability (regarding modification and erasure) of the directories including binaries and related libraries implementing those functions (also in memory after loading). Those elements are, for a regular user, at most read-only, and not executable for some of them.

The security needs for the configuration is handled by a subset of files and directories of the filesystem, with security needs on integrity and availability for most of them and confidentiality for a few ones.

The following table summarizes this security needs on the configuration.

Config file		Description	Sp.	User rights			Group rights			Other rights			Owner /Group	
Path	Filename													
/etc/	group	user group file	-	r	w	-	r	-	-	r	-	-	root	root
/etc/	issue	prelogin message and identification file	-	r	w	-	r	-	-	r	-	-	root	root
/etc/	issue.net	identification file for telnet/SSH sessions	-	r	w	-	r	-	-	r	-	-	root	root
/etc/	login.defs	shadow password suite configuration	-	r	w	-	r	-	-	r	-	-	root	root
/etc/pam.d/	*	Configurations for PAM services	-	r	w	-	r	-	-	r	-	-	root	root
/etc/	passwd	User account database	-	r	w	-	r	-	-	r	-	-	root	root
/etc/	profile	For specifying system wide environment and startup programs; used by all users;	-	r	w	-	r	-	-	r	-	-	root	root
/etc/	rsyslog.conf	rsyslogd configuration file	-	r	w	-	-	-	-	-	-	-	root	root
/etc/	shadow	shadowed password file	-	r	w	-	-	-	-	-	-	-	root	root
/etc/ssh/	ssh_config	OpenSSH SSH client configuration file (for all users)	-	r	w	-	-	-	-	-	-	-	root	root
/etc/ssh/	sshd_config	OpenSSH SSH daemon configuration file	-	r	w	-	-	-	-	-	-	-	root	root
/home/rtos	All directory	Default user directory	-	r	w	x	r	-	x	-	-	-	rtos	rtos

5 THREATS ON THE ASSETS

This chapter describes the threats scenarios on the assets to be protected by the product in the following way:
A threat agent *makes an action* on **the asset**. The assets are sometimes functional as for instance **the behavior of the product** or **the behavior of the product environment**.
.

Threat agents:

This security target uses the following threat agents in the threats definitions:

- Remote attacker
- Local attacker without credentials
- Local/remote attacker with credentials (simple user trying privilege escalation)

5.1 T1. Remote leak

A remote attacker *leaks* **the trusted channel information**.

This information can be for instance user credentials the attacker will use to gain access to the evaluated product.

5.2 T2. Remote attacker intrusion

A remote attacker *modifies with consistency* **the trusted channel information**.

This modification allows for instance to execute commands or binary code on the evaluated product.

5.3 T3. Bypassing the authentication or other access restrictions

A remote or local attacker without credentials *gains access* on **the evaluated product**.

5.4 T4. Gaining not granted access

A local or remote attacker with credentials *gains* **not granted access on the evaluated product**.

5.5 T5. Altering the general product behavior

An attacker *alters* **the general product behavior**.

6 SPECIFICATION OF THE SECURITY FUNCTIONS

6.1 F1. Identification/Authentication

This is a composite function which includes:

- The ability for the RTOS to associate a reliable identity to the current user (i.e; the user cannot do actions on behalf to another user)
- The ability for the RTOS to ensure the authenticity of the current user trying to open a session, based on a password (console/ssh log in) or a public/private key (ssh log in)
- The ability for the RTOS to lock/close session in case of a period of inactivity
- Other hardening properties from [Linux], such as password complexity enforcement.

This security function is based on the GNU/Linux PAM (Pluggable Authentication Module), configured as per the enhanced level of [LINUX], when applicable.

6.2 F2. Trusted channel

Using the cryptographic communication protocol SSH, the RTOS is able to establish secure and trusted communication channel to and from other IT entities.

This security function is based on the OpenSSH server, configured as per [SSH] recommendations, when applicable. The “Annex B - OpenSSH configuration” reminds all the [SSH] recommendations.

6.3 F3. Segregating users and configuration data

The RTOS offers to the users and/or authorized administrators the ability to modify the configuration of the RTOS and a Discretionary Access Control (DAC) which allow owner of named objects to control the access permissions to these objects and enforcing configuration data is accessed only by granted users.

This security function is based on the GNU/Linux access control and configuration, configured as per the enhanced level of [LINUX], when applicable. The “Annex A – Enhanced Level’s Recommendations” reminds all the enhanced level recommendations and those that are not applicable in the RTOS CSPN context.

7 THREATS COVERAGE

The following table shows the coverage of the threats by the security functions:

Threat/SF	F1	F2	F3
T1. Remote leak	X	X	
T2. Remote attacker intrusion	X	X	
T3. Bypassing the authentication or other access restrictions	X	X	
T4. Gaining not granted access			X
T5. Altering the general product behavior	X	X	X

8 NORMATIVE DOCUMENTS AND EVALUATION METHODS

Reference	Title
[NOTE/20]	Règles relatives à la mise en œuvre des évaluations sécuritaires
[NOTE/21]	Méthodologie pour l'évaluation d'une gamme de produits
[CSPN-CER-I-02]	Critères pour l'évaluation en vue d'une Certification de Sécurité de Premier Niveau
[CSPN-NOTE/01]	Méthodologie pour l'évaluation en vue d'une Certification de Sécurité de Premier Niveau - Contenu du RTE
[CSPN-NOTE/21]	Méthodologie pour l'évaluation d'une gamme de produits
[LINUX]	Configuration recommendations of a GNU/Linux System, ANSSI-BP-028-EN
[SSH]	Technical report (Open)SSH secure use recommendations, AT-NT-007-EN/ANSSI/SDE/NP

9 ANNEX A – ENHANCED LEVEL'S RECOMMENDATIONS

The following table reminds the Enhanced Level's recommendations from [LINUX].

Réf	Recommendation
R1	Minimization of installed services
R2	Minimization of configuration
R3	Principle of least privilege
R5	Defense in depth principle
R6	Network services partitioning
R7	Logging of service activity
R8	Regular updates
R9	Hardware configuration
R10	32 and 64 bit architecture
R12	Partitioning type
R13	Access Restrictions on the /boot directory
R14	Installation of packages reduced to the bare necessities
R15	Choice of package repositories
R16	Hardened package repositories
R17	Boot loader password
R18	Robustness of the administrator password
R19	Accountability of administration operations
R20	Installation of secret or trusted elements
R21	Hardening and monitoring of services subject to arbitrary flows
R22	Setting up network sysctl
R23	Setting system sysctl
R24	Disabling the loading of kernel modules
R25	Yama module sysctl configuration
R26	Disabling unused user accounts
R27	Disabling service accounts
R28	Uniqueness and exclusivity of system service accounts
R29	User session timeout
R30	Applications using PAM
R31	Securing PAM Authentication Network Services
R32	Protection of stored passwords
R33	Securing access to remote user databases
R34	Separation of System Accounts and Directory Administrator
R35	umask value
R36	Rights to access sensitive content files
R37	Executable with setuid and setgid bits
R38	Executable setuid root
R39	Temporary directories dedicated to each account
R40	Sticky bit and write access rights

Réf	Recommendation
R41	Securing access for named sockets and pipes
R42	Services and resident daemons in memory
R43	Hardening and configuring the syslog service
R44	Partitioning the syslog service by chroot
R46	Service Activity Logs
R47	Dedicated partition for logs
R48	Configuring the local messaging service
R49	Messaging Aliases for Service Account
R50	Logging activity by auditd
R53	Restricting access of deployed services
R54	Virtualization components hardening
R55	chroot jail and access right for partitioned service
R56	Enablement and usage of chroot by a service
R57	Group dedicated to the use of sudo
R58	Sudo configuration guidelines
R59	User authentication running sudo
R60	Privileges of target sudo users
R61	Limiting the number of commands requiring the use of the EXEC option
R62	Good use of negation in a sudoers file
R63	Explicit arguments in sudo specifications
R64	Good use of sudoedit

Note: Applicable recommendations are covered by:

- the RTOS default configuration,
- the application of the administration and user guides.

The following recommendations are not applicable in the context of the RTOS CSPN:

- R9: The RTOS is software only and the referenced technical note concerns x86 architectures only.
- R33: The RTOS does not use external authentication
- R48 and R49: The RTOS does not include messaging service
- R57 to R64: There is no sudo on the RTOS

10 ANNEX B - OPENSSH CONFIGURATION

The following table reminds the recommendations from [SSH].

Ref.	Recommendation
R1	Only version 2 of the SSH protocol shall be authorized.
R2	SSH shall be used instead of historical protocols (TELNET, RSH, RLOGIN) for remote shell access.
R3	TELNET, RSH and RLOGIN remote access servers shall be uninstalled from the system.
R4	SCP or SFTP shall be used instead of historical protocols (RCP, FTP) for file transfers.
R5	The implementation of SSH tunnels shall only be applied to protocols that do not provide robust security mechanisms and that can benefit from it (for example: X11, VNC). This recommendation does not exempt from using additional low level security protocols, such as IPsec (Refer to "IPsec Security Recommendations" available at www.ssi.gouv.fr for additional information).
R6	The server authenticity shall always be checked prior to access. This is achieved through preliminary machine authentication by checking the server public key fingerprint, or by verifying the server certificate.
R7	The use of DSA keys is not recommended.
R8	The minimum key size shall be 2048 bits for RSA.
R9	The minimum key size shall be 256 bits for ECDSA.
R10	ECDSA keys should be favoured over RSA keys when supported by SSH clients and servers.
R11	Keys should be generated in a context where the RNG is reliable, or at least in an environment where enough entropy has been accumulated.
R12	Some rules can ensure that the entropy pool is properly filled: • keys must be generated on a physical equipment; • system must have several independent sources of entropy; • key generation shall occur only after a long period of activity (several minutes or even hours)
R13	The private key should only be known by the entity who needs to prove its identity to a third party and possibly to a trusted authority. This private key should be properly protected in order to avoid its disclosure to any unauthorized person.
R14	Private keys shall be password protected using AES128-CBC mode.
R15	The encryption algorithm shall either be AES128-CTR, AES192-CTR or AES256-CTR. The integrity mechanism shall rely on HMAC-SHA1, HMAC-SHA256 or HMACSHA512.
R16	A preliminary step in hardening the sshd service is to use proper compilation flags. Refer to “Security recommendation for systems using GNU/Linux” as an example.
R17	User authentication should be performed with one of the following mechanisms, given by order of preference: • ECDSA asymmetric cryptography; • RSA asymmetric cryptography; • symmetric cryptography (Kerberos tickets from the GSSAPI); • authentication modules that expose neither the user password nor its hash (third-party PAM or BSD Auth modules); • password check against a database (such as passwd/shadow) or a directory.
R18	Users rights shall follow the least privilege principle. Restrictions can be applied on several parameters: available commands, source IP, redirection of forwarding permissions, …
R19	When SSH bouncing is necessary through a relay host, Agent Forwarding (-A option of ssh) should be used.
R20	The relay host server shall be a trusted host.
R21	Every user must have his own, unique, non-transferable account.

Ref.	Recommendation
R22	Access to a service shall be restricted to users having a legitimate need. This restriction shall apply on a white-list basis: only explicitly allowed users shall connect to a host via SSH and possibly from specified source IP addresses.
R23	Access to a service shall be restricted to users having a legitimate need. This restriction shall apply on a white-list basis: only explicitly allowed users shall connect to a host via SSH and possibly from specified source IP addresses.
R24	Users shall only execute strictly necessary commands. This restriction can be achieved in the following ways: – using the ForceCommand directive on a per user basis in the sshd_config file; – specifying some options in the authorized_keys file (See 4.3.1); – using secure binaries such as sudo or su
R25	The SSH server shall only listen on the administration network.
R26	When the SSH server is exposed to an uncontrolled network, one should change its listening port (22). Preference should be given to privileged ports (below 1024) to prevent spoofing attempts by unprivileged services on the remote machine. On a controlled network, the SSH server shall only listen on a management network interface, separated from the operational network.
R27	Except for duly justified needs, any flow forwarding feature shall be turned off: • in the SSH server configuration; • in the local firewall by blocking connections.
R28	X11 forwarding shall be disabled on the server.
R29	It is recommended to create distinct CAs when their roles differ. There will be, for example: • one CA for the “hosts” CA role; • one CA for the “users” CA role. Each CA private key shall be protected by a unique and robust password.
R30	If a key cannot be considered safe anymore, it shall be quickly revoked at the SSH level.
R31	SSH host key fingerprints obtained through DNS records should not be trusted without complimentary verifications.

Note: The RTOS default configuration covers some of those recommendations. The other ones are covered by the application of the administration guide.

10.1 Common parameters for ssh_config and ssd_config

The /etc/ssh_config and /etc/sshd_config files should include the following lines:

Protocol 2
Ciphers chacha20-poly1305@openssh.com, aes128-ctr, aes192-ctr, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com
MACs umac-64-etm@openssh.com, umac-128-etm@openssh.com, hmac-sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com
StrictHostKeyChecking ask

10.2 Specific parameters for sshd_config

The /etc/sshd_config file should include the following lines for all remote hosts:

```
PermitRootLogin no
PubkeyAuthentication yes
IgnoreRhosts yes
StrictModes yes
SyslogFacility AUTH
LogLevel VERBOSE
PermitEmptyPasswords no
X11Forwarding no
PrintLastLog yes
HostbasedAuthentication no
UsePAM yes
MaxAuthTries 3
PermitUserEnvironment no
ClientAliveInterval 300
ClientAliveCountMax 0
PasswordAuthentication no
AllowTcpForwarding no
MaxStartups     10:30:100
LoginGraceTime 120
AllowAgentForwarding no
VerifyHostKeyDNS ask
ForwardAgent no
ForwardX11 no
IgnoreUserKnownHosts yes
```

For remote hosts allowed to transfert files with sftp, the following line can be added:

```
Subsystem sftp internal-sftp -l INFO
```