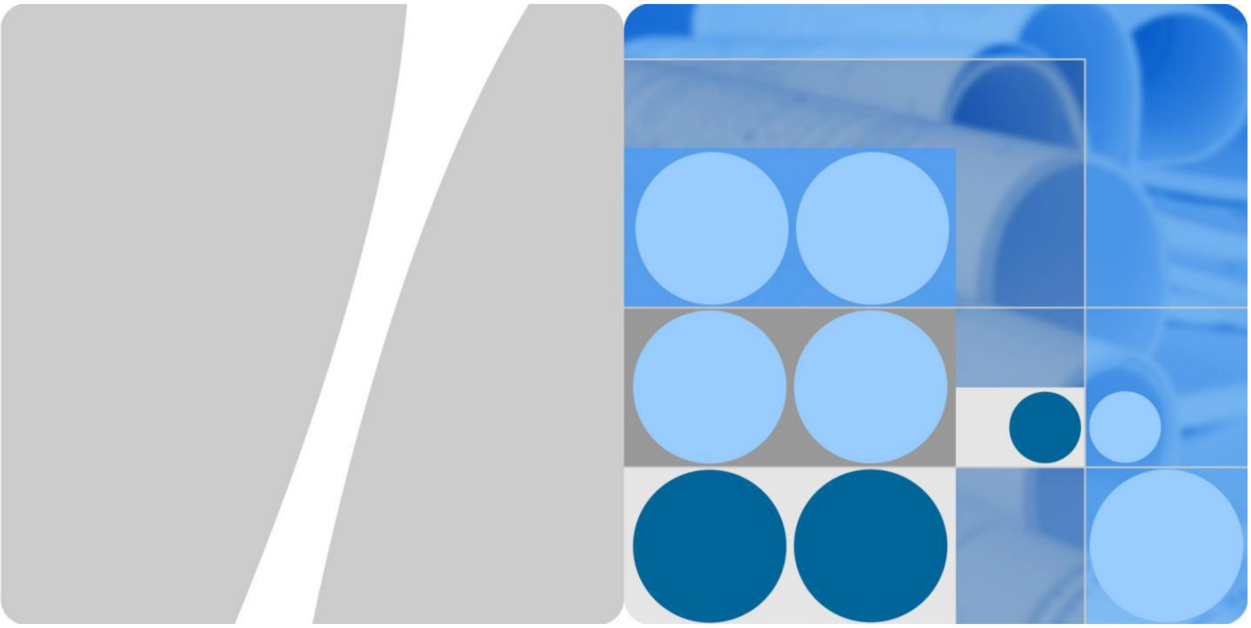


Huawei EulerOS Version 2.0

Common Criteria Evaluation

Security Target

(Against NIAP PP)

Version	0.9
Status	Released
Last update	2017-12-20
Classification	Public

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://e.huawei.com

About This Document

Purpose

This document provides description about ST (Security Target).

Change History

Changes between document issues are cumulative. The latest document issue contains all the changes made in earlier issues.

Date	Revision Version	Section Number	Change Description	Author
2017-05-17	0.1	ALL	Initial Draft	EulerOS team
2017-07-29	0.2	ALL	Apply Typographical conventions for all SFR	EulerOS team
2017-09-13	0.3	All	RSA 4096 removed; SSH packets number specification before rekeying is removed; Max ssh packet size changed to 256KB; Adjusted based on all TDs related to 'PP_OS_v4.1';	EulerOS team

2017-09-18	0.4	ALL	Change according to EE’s feedback in 2017/9/14 22:27: Openssl updated to 1.0.2k to support hostname verification; Uniqueness of counter for aes128-ctr and aes256-ctr is briefed in section 6.1.2; Sensitive persistent data is collected in section 6.1.3; Section 6.6.6 is revised, giving algorithm for certificate validation.	

2017-09-28	0.5	ALL	Explanation of ‘/etc/passwd NOT being credential file’ in TSS;
			“OCSP stapling” removed;
				EulerOS team

2017-10-10	0.6	ALL	Complied TDs are added in section “Conformance Claims”;
			Update section ‘key destruction’ (FCS_CKM.4) based on TD0239;
			Some config items are removed from the table in section “FMT_SMF_EXT.1 Extended: Specification of Management Functions”;
			Authentication method (only for SSH) added;
			Remove rekey condition “*1 Gigabyte*” in FCS_SSHS_EXT.1.7;
			Max packet size in SSH is changed from 35000 t0 256K (for SSH2);
				EulerOS team

2017-11-03	0.7	FCS_CKM.4	Add selection of “removal of power to the memory”. Section 6.1.1 also revised accordingly.	EulerOS team
2017-11-30	0.8	6.4.4 Platform Integrity and Code Integrity	Remove ‘IMA-appraisal’ from boot integrity.	EulerOS team

		FCS_CK.4	Leaving ‘poweroff’ as the only method for cryptographic key destruction;	
		FPT_TST_EXT.1	Addition of ‘codesigning check’;	
2017-12-20	0.9	TSS-protection from implementation weakness	compiler options summarized;	EulerOS team
		Section 6.4.5	A new subsection ‘3. For security update’ added.	

Contents

About This Document 4

Contents 7

List of Tables 11

1. Security Target Introduction 12

1.1 Security Target Reference 12

1.2 TOE Reference 12

1.3 TOE Overview 12

1.3.1 TOE Type 13

1.3.2 Major Security Features 13

1.3.3 Non-TOE Hardware/Software/Firmware supported 15

1.3.4 Intended Method of Use 15

1.4 TOE Description 16

1.4.1 Evaluated Configuration 16

1.4.2 TOE boundaries 16

2. CC Conformance Claims 19

3. Security Problem Definition 20

3.1 Threats 20

3.2 Assumptions 20

3.3 Organizational Security Policies 21

4. Security Objectives 22

4.1 Security Objectives for the TOE 22

4.2 Security Objectives for the Operational Environment 23

5. Security Requirements 24

5.1 TOE Security Functional Requirements 25

5.1.1 Cryptographic Support (FCS) 26

5.1.2 User Data Protection (FDP) 33

5.1.3 Security Management (FMT) 33

5.1.4 Protection of the TSF (FPT) 36

5.1.5 Audit Data Generation (FAU) 38

5.1.6 Identification and Authentication (FIA) 39

5.1.7 Trusted Path/Channels (FTP) 40

5.2 TOE Security Assurance Requirements 41

5.3 Security Requirements Rationale 43

5.3.1 Security Functional Requirements Rationale 43

5.3.2 Security Assurance Requirements Rationale 49

5.3.3 Rationale for Conformance to Protection Profile 49

6 TOE Summary Specification (TSS) 50

6.1 Cryptographic Support 50

6.1.1 Cryptographic Algorithms and Operations 50

6.1.2 Cryptographic network services 56

6.1.3 Data Protection 61

6.1.4 SFR Summary 62

6.2 User data protection 63

6.2.1 Discretionary Access Control 63

6.2.2 VPN client 66

6.2.3 SFR Summary 67

6.3 Security Management 67

6.3.1 SFR Summary 69

6.4 Protection of the TSF 70

6.4.1 Separation and Domain Isolation 70

6.4.2 Protection of OS Binaries, Audit and Configuration Data 71

6.4.3 Protection from Implementation Weaknesses 71

6.4.4 Platform Integrity and Code Integrity 72

6.4.5 OS and Application Updates 74

6.4.6 SFR summary 79

6.5 Audit 79

6.5.1 Audit event selection 80

6.5.2 Audit trail 80

6.5.3 Audit log access protection 81

6.5.4 SFR summary 81

6.6 Identification and Authentication 81

6.6.1 PAM-based identification and authentication mechanisms 82

6.6.2 User Identity Changing 83

6.6.3 Authentication Data Management 84

6.6.4 SSH key-based authentication 85

6.6.5 Session locking 85

6.6.6 X.509 Certificate Validation and Generation 85

6.6.7 SFR Summary 87

6.7 Trusted Path/Channels 87

6.7.1 Local trusted path 87

6.7.2 Network-based trusted channel 88

6.7.3 SFR Summary 89

7 Appendix A. SBOP files and rationale. 90

A. Acronyms 107

List of Tables

Table 1 Security Functional Requirements 26

Table 2 TOE Security Assurance Requirements 42

Table 3 Cryptographic Algorithm Standards supported by EulerOS 53

Table 4 Origin/storage/zeroization of keys 55

Table 5 TLS RFCs implemented in EulerOS 58

1. Security Target Introduction

This section presents the following information required for a Common Criteria (CC) evaluation:

- Identifies the Security Target (ST) and the Target of Evaluation (TOE)
- Specifies the security target conventions
- Describes the organization of the security target

1.1 Security Target Reference

Name:	EulerOS 2.0 Security Target
Version:	0.9
Publication Date:	2017-12-20
Author:	Huawei Technologies Co., Ltd.

1.2 TOE Reference

Name:	EulerOS
Version:	2.0
Build	3.10.0-327.59.59.46.h34.x86_64
Release Date	Dec 18 2017

1.3 TOE Overview

The TOE, EulerOS V2.0, is a general purpose, multi-user, multi-tasking Linux based operating system. It provides a platform for a variety of applications, including services for cloud environments.

TOE 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 TOE Security Functions (TSFs) consist of functions of EulerOS 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.

The TOE includes standard networking applications, such as *sshd(8)*, which allow to access the TOE via cryptographically protected communication channel.

1.3.1 TOE Type

The TOE type is a Linux-based general-purpose operating system, supporting preemptive multitasking, multiprocessor, and multi-user.

1.3.2 Major Security Features

The primary security features of the TOE include:

- **Cryptographic communication:** The TOE provides cryptographic secured communication to either allow remote entities to log into the TOE or local used to establish secure communications. The SSHv2 protocol is provided to set up interactive session with the TOE. The TOE provides both the server side and the client side applications. Using the OpenSSH suite, password-based and public-key-based authentication are allowed. The TOE provides the capability of configure a VPN channel for a cryptographically secured communication with other remote entities. The TOE implements TLS protocol to enable a trusted network path that is used for client and server authentication, as well as HTTPS.
- **Encrypted user data storage:** EulerOS provides data protection APIs in *openssl* package (eg, *EVP_EncryptInit_ex*, *EVP_EncryptUpdate*, *EVP_EncryptFinal_ex*), which applications can use to protect any persisted data that the developer deems to be sensitive.
- **Auditing**: The Lightweight Audit Framework (LAF) is designed to be an audit system making EulerOS compliant with the requirements from Common Criteria. LAF is able to intercept all system calls as well as retrieving audit log entries from privileged user space applications. The subsystem allows to configure the events to be actually audited

from the set of all events that are possible to be audited, and to review and search audit logs retrieved.

- **Identification and Authentication**: Each user accessing the TOE is identified by a name, and is authenticated based on a password. PAM (pluggable authentication module) mechanism can be used to define or configure authentication policy, session management, password update and so on.
- **Data Protection**: Discretionary Access Control (DAC)allows owners of named objects to control the access permissions to these objects. The owners can permit or deny access by other users based on the configured permission settings. The DAC mechanism is also used to ensure that untrusted users cannot tamper with the TOE mechanisms.
- Runtime Protection mechanisms: The TOE provides mechanisms to prevent, or significantly increase the complexity of, exploitation of common buffer overflow and similar attacks. These mechanisms are used for the TSF and are also available to untrusted code. The TOE implements multiple countermeasures against exploitation of programming errors. Classical programming errors, such as buffer overflows, are exploitable using a set of exploitation techniques. The TOE blocks or significantly increases the challenge to use these techniques with the following different approaches:
    - ➢ Prevention of code execution on stack. This prevents buffer overflow attacks which writes executable code (e.g. the shellcode) into a stack variable and causes the CPU to execute it.
    - ➢ Address space layout randomization (ASLR), a security technique also involved in protection from buffer overflow attacks. In order to prevent an attacker from reliably jumping to, for example, a particular exploited function in memory, ASLR randomly arranges the address space positions of key data areas of a process, including the base of the executable and the positions of the stack, heap and libraries.
    - ➢ Boot integrity and system integrity. All components in the boot chain are measured at boot time, and some key system files are also measured at system uptime. All the measurements can be verified later by the administrator to see if some files are compromised. All updated packages are authenticated and

verified based on their signature to ensure the integrity of the whole system.

- **Security Management:** The security management facilities provided by the TOE are usable by authorized users and/or authorized administrators to modify the configuration of TSF. The TOE allows remote management via OpenSSH. Administrative users can log in remotely and perform the same management tasks as a locally operating administrator.
- **Trusted Path for Communications:** EulerOS uses HTTPS, TLS and SSH to provide a trusted path for communications.

1.3.3 Non-TOE Hardware/Software/Firmware supported

Non-TOE Hardware Identification: The following physical and virtual hardware platforms, corresponding firmware, and components are supported by the TOE :

- FusionCube 6000 and 6000C (with TPM chip embedded)
- FusionServer RH2288H V3 Rack Server (with TPM chip embedded)
- FusionServer RH8100 V3 Rack Server (with TPM chip embedded)
- FusionServer X6800 Data Center Server (with TPM chip embedded)
- FusionServer XH628 V3 Server Node (with TPM chip embedded)
- Linux QEMU-KVM-1.5.3 virtual platform (with virtualized TPM chip)

Note: the TPM chip used in evaluation is Infineon SLB9665.

1.3.4 Intended Method of Use

1.3.4.1 General-purpose computing environment

The TOE 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 gets access to a general computing environment, allowing launching user applications, issuing user commands at shell level, creating and accessing files. The TOE provides adequate mechanisms to separate the users and protect their data. Privileged commands are restricted to only administrative users.

The TOE operates in a networked environment with other instantiates of the TOE as well as other well-behaved peer systems.

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

The TOE 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 TOE.

The TOE 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 supplemental 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.

1.4 TOE Description

1.4.1 Evaluated Configuration

The TOE was evaluated on the following physical platforms:

- FusionServer RH2288H V3 Rack Server (with TPM chip embedded).

Note: the TPM chip used is Infineon SLB9665.

The user needs to follow the instructions defined in the guidance for reaching evaluated configuration.

1.4.2 TOE boundaries

1.4.2.1 Physical boundary

The TOE and its documentation (pdf format) are supplied in two forms: DVD disks, and ISO images distributed via the Huawei Network.

The TOE was evaluated on the following physical platforms:

- FusionServer RH2288H V3 Rack Server (with TPM chip embedded).

The following documentations are provided for the TOE:

- Huawei EulerOS V2.0 Installation Guide, Version 0.2
- Huawei EulerOS V2.0 User Guide, Version 0.2

1.4.2.2 Logical boundary

Conceptually the TOE can be thought of as a collection of the following security services which the security target describes with increasing detail in the remainder of this document:

- Cryptographic Support
- User Data Protection
- Security Management
- Protection of the TOE Security Functions
- Security Audit
- Identification and Authentication
- Trusted Path and Channels

The following security functions are included in the TOE:

- Cryptographic support: The TOE provides full-functional cryptography used in protecting local system/user data and network traffic.
- User data protection: traditional discretionary access control (DAC) is used to allow owners of local named objects to control the access permissions to these objects.
- Security Management: The security management facilities provided by the TOE are only usable by authorized users and/or authorized administrators to modify the configuration of TSF.
- Protection of the TOE Security Functions: All the resources used by TSFs in the TOE are protected by access control. All native binaries are built in a way to avoid buffer overflow, and process address space layout is randomized to hinder attacks. Secure boot and integrity verification are used to keep the whole system from contamination.

- Auditing: the audit subsystem can be configured to intercept all system calls and the stored audit log is protected by access control mechanism.
- Identification and Authentication: User must log in to the TOE before accessing resources on it. PAM is used to define the user password strength and user login behavior.
- Trusted path/channel: The TOE provides facilities to build trusted channels for users to access the TOE remotely and securely.

2. CC Conformance Claims

This TOE and ST are consistent with the following specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 3.1, Revision 5, April 2017, extended (Part 2 extended)
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements Version 3.1, Revision 5 April 2017, (Part 3 extended)
- General Purpose Operating Systems Protection Profile, Version 4.1, March 9, 2016 (GP OS PP)
- Extended Package for Secure Shell (SSH), Version 1.0, February 2, 2016

The security functional requirements and assurance activities have been modified with the following NIAP Technical Decisions (TDs):

- 0246 – Assurance Activity for FIA_UAU.5.2
- 0244 – FCS_TLSC_EXT - TLS Client Curves Allowed
- 0243 – SSH Key-Based Authentication
- 0239 – Cryptographic Key Destruction in OS PP
- 0208 – Remote Users in OSPP
- 0163 – Update to FCS_TLSC_EXT.1.1 Test 5.4 and FCS_TLSS_EXT.1.1 Test
- 0107 - FCS_CKM - ANSI X9.31-1998, Section 4.1 for Cryptographic Key Generation.
- 0104 – FMT_SMF and FMT_MOF in OS PP

3. Security Problem Definition

The security problem definition consists of the threats to security, organizational security policies, and usage assumptions as they relate to the TOE. The assumptions, threats, and policies are copied from the General Purpose Operating Systems Protection Profile, Version 4.1, March 9, 2016 (“GP OS PP”).

3.1 Threats

T.NETWORK_ATTACK

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with applications and services running on or part of the OS with the intent of compromise. Engagement may consist of altering existing legitimate communications.

T.NETWORK_EAVESDROP

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between applications and services that are running on or part of the OS.

T.LOCAL_ATTACK

An attacker may compromise applications running on the OS. The compromised application may provide maliciously formatted input to the OS through a variety of channels including unprivileged system calls and messaging via the file system.

T.LIMITED_PHYSICAL_ACCESS

An attacker may attempt to access data on the OS while having a limited amount of time with the physical device.

3.2 Assumptions

A.PLATFORM

The OS relies upon a trustworthy computing platform for its execution. This underlying platform is out of scope of this PP.

A.PROPER_USER

The user of the OS is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy. At the same time, malicious software could act *as* the user, so requirements which confine malicious subjects are still in scope.

A.PROPER_ADMIN

The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

3.3 Organizational Security Policies

There are no Organizational Security Policies for the protection profile.

4. Security Objectives

This section defines the security objectives of EulerOS and its supporting environment. Security objectives, categorized as either TOE security objectives or objectives by the supporting environment, reflect the stated intent to counter identified threats, comply with any organizational security policies identified, or address identified assumptions. All of the identified threats, organizational policies, and assumptions are addressed under one of the categories below.

4.1 Security Objectives for the TOE

Below are the security objectives for EulerOS, which are needed to comply with the GP OS PP.

O.ACCOUNTABILITY

Conformant OSs ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the operating system and discover its cause. Gathering event information and immediately transmitting it to another system can also enable incident response in the event of system compromise.

O.INTEGRITY

Conformant OSs ensure the integrity of their update packages. OSs are seldom if ever shipped without errors, and the ability to deploy patches and updates with integrity is critical to enterprise network security. Conformant OSs provide execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems.

O.MANAGEMENT

To facilitate management by users and the enterprise, conformant OSes provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration and application execution control.

O.PROTECTED_STORAGE

To address the issue of loss of confidentiality of credentials in the event of loss of physical control of the storage medium, conformant OSs provide data-at-rest protection for credentials. Conformant OSes also provide access controls which allow users to keep their files private from other users of the same system.

O.PROTECTED_COMMS

To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant OSs provide mechanisms to create trusted channels for CSP and sensitive data. Both CSP and sensitive data should not be exposed outside of the platform.

4.2 Security Objectives for the Operational Environment

The TOE is assumed to be complete and self-contained and, as such, is not dependent upon any other products to perform properly. However, certain objectives with respect to the general operating environment must be met. Below are the security objectives for the operational environment as specified in the protection profile.

OE.PLATFORM

The OS relies on being installed on trusted hardware.

OE.PROPER_USER

The user of the OS is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy. Standard user accounts are provisioned in accordance with the least privilege model. Users requiring higher levels of access should have a separate account dedicated for that use.

OE.PROPER_ADMIN

The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

5. Security Requirements

The section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) for the TOE. The requirements in this section have been drawn from the General Purpose Operating Systems Protection Profile, Version 4.1, March 9, 2016 (GP OS PP), the Common Criteria, or are defined in the following section.

Conventions:

Where requirements are drawn from the protection profile, the requirements are copied verbatim, except for some changes to required identifiers to match the iteration convention of this document, from that protection profile and only operations performed in this security target are identified.

The extended requirements, extended component definitions and extended requirement conventions in this security target are drawn from the protection profile; the security target reuses the conventions from the protection profile which include the use of the word “Extended” and the “_EXT” identifier to denote extended functional requirements. The security target assumes that the protection profile correctly defines the extended components and so they are not reproduced in the security target.

The following conventions are used to identify operations:

> **Refinement:** Refinements are identified using bold text (e.g., **added text**) for additions and strike-through text (e.g., ~~deleted text~~) for deletions.
> **Selection** (denoted by *italicized text*, **bold** and in square brackets): is used to select one or more options provided by the [CC] in stating a requirement.
> **Assignment** operation (denoted by *italicized text* in square brackets): is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
> **Iteration** operation: are identified with either a number or element inside parentheses (e.g. "(1)")

5.1 TOE Security Functional Requirements

Below are the Security Functional Requirements for the TOE.

Requirement Class	Requirement Component
Security Audit (FAU)	Audit Data Generation (FAU_GEN.1)
Cryptographic Support (FCS)	Cryptographic Key Generation for (FCS_CKM.1(1))
	Cryptographic Key Establishment (FCS_CKM.2(1))
	Cryptographic Key Destruction (FCS_CKM.4)
	Cryptographic Operation for Data Encryption/Decryption (FCS_COP.1(1))
	Cryptographic Operation for Data Encryption/Decryption (FCS_COP.1(SSH))
	Cryptographic Operation for Hashing (FCS_COP.1(2))
	Cryptographic Operation for Signing (FCS_COP.1(3))
	Cryptographic Operation for Keyed Hash Algorithms (FCS_COP.1(4))
	Random Bit Generation (FCS_RBG_EXT.1)
	Storage of Sensitive Data (FCS_STO_EXT.1)
	TLS Client Protocol (FCS_TLSC_EXT.1)
	TLS Client Protocol (FCS_TLSC_EXT.2)
	SSH Protocol (FCS_SSH_EXT.1)
	SSH Protocol – Client (FCS_SSHC_EXT.1)
	SSH Protocol - Server (FCS_SSHS_EXT.1)
User Data Protection (FDP)	Access Controls for Protecting User Data (FDP_ACF_EXT.1)
	Information Flow Control (FDP_IFC_EXT.1)
Identification & Authentication (FIA)	Authorization Failure Handling (FIA_AFL.1)
	Multiple Authentication Mechanisms (FIA_UAU.5)
	X.509 Certification Validation (FIA_X509_EXT.1)
	X.509 Certificate Authentication (FIA_X509_EXT.2)
Security Management (FMT)	Management of Security Functions Behavior (FMT_MOF_EXT.1)
	Specification of Management Functions (FMT_SMF_EXT.1 Extended)
Protection of the TSF (FPT)	Access Controls (FPT_ACF_EXT.1)
	Address Space Layout Randomization (FPT_ASLR_EXT.1)
	Stack Buffer Overflow Protection (FPT_SBOP_EXT.1)
	Boot Integrity (FPT_TST_EXT.1)
	Trusted Update (FPT_TUD_EXT.1)

	Trusted Update for Application Software (FPT_TUD_EXT.2)
Trusted Path/Channels (FTP)	Trusted Path (FTP_TRP.1)
	Trusted Channel Communication (FTP_ITC_EXT.1)

Table 1 Security Functional Requirements

5.1.1 Cryptographic Support (FCS)

FCS_CKM.1(1) Cryptographic Key Generation

FCS_CKM.1.1(1)
The OS shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [**selection**:

> ***RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: [selection: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3] ,***
>
> ***ECC schemes using “NIST curves” P-256, P-384 and [selection: P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4***

]

FCS_CKM.2(1) Cryptographic Key Establishment

FCS_CKM.2.1(1)
The OS shall implement functionality to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:
RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, “Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography”
and [**selection**:

> ***Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”***

]

FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1
The TSF shall destroy cryptographic keys in accordance with the specified cryptographic key destruction methods [**selection**:

> ***- For volatile memory, the destruction shall be executed by a*** [**selection**: ***removal of power to the memory***]

]

FCS_COP.1(1) Cryptographic Operation - Encryption/Decryption

FCS_COP.1.1(1)
The OS shall perform encryption/decryption services for data in accordance with a specified cryptographic algorithm [**selection**:

> ***AES-XTS (as defined in NIST SP 800-38E),***
>
> ***AES-CBC (as defined in NIST SP 800-38A)*]**

and [**selection**:

> ***AES Key Wrap (KW) (as defined in NIST SP 800-38F),***
>
> ***AES-GCM (as defined in NIST SP 800-38D)***

] and cryptographic key sizes [**selection**: ***128-bit, 256-bit*]**.

FCS_COP.1(SSH) Cryptographic Operation - Encryption/Decryption

Application Note: *FCS_COP.1(SSH) corresponds to FCS_COP.1(1) in the Extended Package for Secure Shell (SSH) protection profile.*

FCS_COP.1.1(SSH)
The SSH software shall perform encryption/decryption services for data in accordance with a specified cryptographic algorithm AES-CTR (as defined in NIST SP 800-38A) mode and cryptographic key sizes [**selection: *128-bit, 256-bit***].

FCS_COP.1(2) Cryptographic Operation - Hashing

FCS_COP.1.1(2)
The OS shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm SHA-1 and [**selection**:

> ***SHA-256,***
>
> ***SHA-384,***
>
> ***SHA-512***

] and message digest sizes 160 bits and [**selection**:

256 bits,

384 bits,

512 bits

] that meet the following: FIPS Pub 180-4.

FCS_COP.1(3) Cryptographic Operation - Signing

FCS_COP.1.1(3)
The OS shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [**selection**:

RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4,

ECDSA schemes using "NIST curves" P-256, P-384 and [selection: P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5

].

FCS_COP.1(4) Cryptographic Operation - Keyed-Hash Message Authentication

FCS_COP.1.1(4)
The OS shall perform keyed-hash message authentication services in accordance with a specified cryptographic algorithm [**selection**:

SHA-1,

SHA-256,

SHA-384,

SHA-512

] with key sizes [**assignment:** *512 and 1024 bits*] and message digest sizes **[selection: *160 bits, 256 bits, 384 bits, 512 bits*]** that meet the following: FIPS Pub 198-1 *The Keyed-Hash Message Authentication Code* and FIPS Pub 180-4 *Secure Hash Standard*.

FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1
The OS shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [**selection**:

Hash_DRBG (any),

HMAC_DRBG (any),

CTR_DRBG (AES)

]
Application Note:

1) The underlying cryptographic primitives for Hash_DRBG and HMAC_DRBG is SHA-256.

2) The underlying cryptographic primitives for CTR_DRBG is AES-256.

FCS_RBG_EXT.1.2
The deterministic RBG used by the OS shall be seeded by an entropy source that accumulates entropy from a [**selection**:

software-based noise source,

platform-based noise source

] with a minimum of [**selection**:

128 bits

] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

FCS_STO_EXT.1 Storage of Sensitive Data

FCS_STO_EXT.1.1
The OS shall implement functionality to encrypt sensitive data stored in non-volatile storage and provide interfaces to applications to invoke this functionality.

FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1
The OS shall implement TLS 1.2 (RFC 5246) supporting the following cipher suites:
Mandatory cipher suites: TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
Optional cipher suites: [**selection**:

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,

TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,

TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,

TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246

]

FCS_TLSC_EXT.1.2
The OS shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3
The OS shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.2 TLS Client Protocol

FCS_TLSC_EXT.2.1
The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [**selection: *secp256r1, secp384r1, secp521r1***] ~~and no other curves.~~

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1
The SSH software shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254 and [**selection**: ***5647, 5656, 6668***] as a [**selection**: ***client, server***].

FCS_SSHC_EXT.1 SSH Protocol – Client

FCS_SSHC_EXT.1.1
The SSH client shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, and [**selection**: ***password-based***].

FCS_SSHC_EXT.1.2
The SSH client shall ensure that, as described in RFC 4253, packets greater than [**assignment**: *256K*] bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.3
The SSH software shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-ctr, aes256-ctr, [**selection**: ***aes128-cbc, aes256-cbc, AEAD_AES_128_GCM, AEAD_AES_256_GCM***].

FCS_SSHC_EXT.1.4
The SSH client shall ensure that the SSH transport implementation uses [**selection**: ***ssh-rsa, ecdsa-sha2-nistp256***] and [**selection**: ***ecdsa-sha2-nistp384***] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHC_EXT.1.5
The SSH client shall ensure that the SSH transport implementation uses [**selection**: ***hmac-sha1, hmac-sha2-256, hmac-sha2-512***] and

[**selection**: **AEAD_AES_128_GCM, AEAD_AES_256_GCM**] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.6
The SSH client shall ensure that [**selection**: ***ecdh-sha2-nistp256***] and [**selection**: ***ecdh-sha2-nistp384,ecdh-sha2-nistp521***] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.7
The SSH server shall ensure that the SSH connection be rekeyed after [**selection**: ***no more than 1 Gigabyte of data has been transmitted, no more than 1 hour***] using that key.

FCS_SSHC_EXT.1.8
The SSH client shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key or [**selection**: ***no other methods***] as described in RFC 4251 section 4.1.

FCS_SSHS_EXT.1 SSH Protocol - Server

FCS_SSHS_EXT.1.1
The SSH server shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, and [**selection**: ***password-based***].

FCS_SSHS_EXT.1.2
The SSH server shall ensure that, as described in RFC 4253, packets greater than [**assignment:** *256K*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.3
The SSH server shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-ctr, aes256-ctr, [**selection**: ***aes128-cbc, aes256-cbc, AEAD_AES_128_GCM, AEAD_AES_256_GCM***].

FCS_SSHS_EXT.1.4
The SSH server shall ensure that the SSH transport implementation uses [**selection**: ***ssh-rsa, ecdsa-sha2-nistp256***] and [**selection**: ***ecdsa-sha2-***

nistp384] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.5
The SSH server shall ensure that the SSH transport implementation uses [**selection: *hmac-sha1, hmac-sha2-256, hmac-sha2-512***] and [**selection: *AEAD_AES_128_GCM, AEAD_AES_256_GCM***] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.6

The SSH server shall ensure that [**selection: *ecdh-sha2-nistp256***] and [**selection: *ecdh-sha2-nistp384, ecdh-sha2-nistp521***] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.7
The SSH server shall ensure that the SSH connection be rekeyed after [**selection: *no more than 1 hour***] using that key.

5.1.2 User Data Protection (FDP)

FDP_ACF_EXT.1 Access Controls for Protecting User Data

FDP_ACF_EXT.1.1
The OS shall implement access controls which can prohibit unprivileged users from accessing files and directories owned by other users.

FDP_IFC_EXT.1 Information flow control

FDP_IFC_EXT.1.1
The OS shall [**selection**:

> ***provide an interface which allows a VPN client to protect all IP traffic using IPsec***

] with the exception of IP traffic required to establish the VPN connection.

5.1.3 Security Management (FMT)

FMT_MOF_EXT.1 Extended: Management of security functions behavior

FMT_MOF_EXT.1.1
The TSF shall restrict the ability to perform the function indicated in column 3 of the “Management Functions” table in FMT_SMF_EXT.1.1 to the administrator.

FMT_SMF_EXT.1 Extended: Specification of Management Functions

FMT_SMF_EXT.1.1 The TSF shall be capable of performing the following management functions:

Management Function	FMT_SMF_EXT.1	FMT_MOF_EXT.1
Enable/disable screen lock	M	O
Configure screen lock inactivity timeout	M	O
Configure local audit storage capacity	M	X
Configure minimum password Length	O	X
Configure minimum number of special characters in password	O	X
Configure minimum number of numeric characters in password	O	X
Configure minimum number of uppercase characters in password	O	X

Configure minimum number of lowercase characters in password	O	X
Configure remote connection inactivity timeout	O	X
~~enable/disable unauthenticated logon~~	~~O~~	~~M~~
Configure lockout policy for unsuccessful authentication attempts through [**selection:** ***timeouts between attempts, limiting number of attempts during a time period***]	O	X
Configure host-based firewall	O	X
Configure name/address of audit/logging server to which to send audit/logging records	O	X
Configure audit rules	O	X
Configure name/address of network time server	O	O
~~Enable/disable automatic software update~~	~~O~~	~~O~~
~~Configure WiFi interface~~	~~O~~	~~O~~

~~Enable/disable Bluetooth interface~~	~~O~~	~~O~~
Configure USB interfaces	O	X
~~Enable/disable [**assignment**: list of other external interfaces]~~	~~O~~	~~O~~
[**assignment**:*none]*	O	O

Application Note: *The intent of this requirement is to ensure that the ST is populated with the management functions that are provided by the TOE. This enables developers of compliance checklists, including those provided as operational user guidance, to leverage this table by providing enterprise-specific values for each evaluated item.*
Functions with strikethrough means that the TOE does NOT support the operation.

5.1.4 Protection of the TSF (FPT)

FPT_ACF_EXT.1 Access controls

FPT_ACF_EXT.1.1
The OS shall implement access controls which prohibit unprivileged users from modifying:

Kernel and its drivers/modules
Security audit logs
Shared libraries
System executables
System configuration files
[**assignment**: *none]*

FPT_ACF_EXT.1.2
The OS shall implement access controls which prohibit unprivileged users from reading:

Security audit logs
System-wide credential repositories
[**assignment**: *none]*

FPT_ASLR_EXT.1 Address Space Layout Randomization

FPT_ASLR_EXT.1.1
The OS shall always randomize process address space memory locations except for [**assignment**: *none]*.

FPT_SBOP_EXT.1 Stack Buffer Overflow Protection

FPT_SBOP_EXT.1.1
The OS shall be compiled with stack-based buffer overflow protections enabled.

FPT_TST_EXT.1 Boot Integrity

FPT_TST_EXT.1.1
The OS shall verify the integrity of the bootchain up through the OS kernel and [**selection**:

> *[**assignment**: operating system executable code and application executable code]*

] prior to its execution through the use of [**selection**:

> ***a digital signature using a hardware-protected asymmetric key***

]

FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1
The OS shall provide the ability to check for updates to the OS software itself.

FPT_TUD_EXT.1.2
The OS shall cryptographically verify updates to itself using a digital signature prior to installation using schemes specified in FCS_COP.1(3).

FPT_TUD_EXT.2 Trusted Update for Application Software

FPT_TUD_EXT.2.1
The OS shall provide the ability to check for updates to application software.

FPT_TUD_EXT.2.2
The OS shall cryptographically verify the integrity of updates to applications using a digital signature specified by FCS_COP.1(3) prior to installation.

5.1.5 Audit Data Generation (FAU)

FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1
The OS shall be able to generate an audit record of the following auditable events:

a. Start-up and shut-down of the audit functions;
b. All auditable events for the not specified level of audit; and
c.
   - Authentication events (Success/Failure);
   - Use of privileged/special rights events (Successful and unsuccessful security, audit, and configuration changes);
   - Privilege or role escalation events (Success/Failure);
   - [**selection**:

     File and object events (Successful and unsuccessful attempts to create, access, delete, modify, modify permissions),

     User and Group management events (Successful and unsuccessful add, delete, modify, disable),

     Audit and log data access events (Success/Failure),

     Kernel module loading and unloading events (Success/Failure),

     Administrator or root-level access events (Success/Failure),

     *[**assignment**: none].*

     ]

FAU_GEN.1.2
The OS shall record within each audit record at least the following information:

a. Date and time of the event, type of event, subject identity (if applicable), and outcome (success or failure) of the event; and
b. For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**assignment**: *none]*.

5.1.6 Identification and Authentication (FIA)

FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1
The OS shall detect when [**selection**:

> *[**assignment**: an administrator configurable positive integer within [**assignment**: greater than or equal to 3]]*

] unsuccessful authentication attempts for [**selection**:

> ***authentication based on user name and password***

] occur related to [**assignment**: *the last successful authentication by that user within 300 seconds]*.

FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts for an account has been met, the OS shall: [**selection**: *Account Lockout*]

FIA_UAU.5 Multiple Authentication Mechanisms

FIA_UAU.5.1
The OS shall provide the following authentication mechanisms [**selection**:

> ***authentication based on user name and password,***
>
> ***for use in SSH only SSH public key-based authentication as specified by the Extended Package for Secure Shell***

] to support user authentication.

FIA_UAU.5.2
The OS shall authenticate any user's claimed identity according to the [**assignment**: *authentication based on username and password is performed for TOE-originated requests and with credentials stored by the OS, Remote authentication for SSH based on username and password or public key-based]*.

FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1
The OS shall implement functionality to validate certificates in accordance with the following rules:

RFC 5280 certificate validation and certificate path validation.

The certificate path must terminate with a trusted CA certificate.
The OS shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
The OS shall validate the revocation status of the certificate using [**selection**: ***a Certificate Revocation List (CRL) as specified in RFC 5759***].
The OS shall validate the extendedKeyUsage field according to the following rules:

- Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
- Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
- Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
- S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the extendedKeyUsage field.
- OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
- (Conditional) Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field.

FIA_X509_EXT.1.2
The OS shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1
The OS shall use X.509v3 certificates as defined by RFC 5280 to support authentication for TLS and [**selection**: ***HTTPS***] connections.

5.1.7 Trusted Path/Channels (FTP)

FTP_ITC_EXT.1 Trusted channel communication

FTP_ITC_EXT.1.1
The OS shall use [**selection**:

> ***TLS as conforming to** **FCS_TLSC_EXT.1,***
>
> ***SSH as conforming to the** **Extended Package for Secure Shell***

] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: [**selection**: ***management server***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_TRP.1 Trusted Path

FTP_TRP.1.1
The OS shall provide a communication path between itself and [**selection**: ***remote, local***] users that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from modification and disclosure.

FTP_TRP.1.2
The OS shall permit [**selection**: ***the TSF*, *local users*, *remote users***] to initiate communication via the trusted path.

FTP_TRP.1.3
The OS shall require use of the trusted path for all remote administrative actions.

5.2 TOE Security Assurance Requirements

This table below gives the set of SARs from CC part 3 that are required in evaluations against the General Purpose Operating Systems Protection Profile.

Requirement Class	Requirement Component
Security Target (ASE)	ST Introduction (ASE_INT.1)
	Conformance Claims (ASE_CCL.1)
	Security Objectives (ASE_OBJ.1)
	Extended Components Definition

	(ASE_ECD.1)
	Stated Security Requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE Summary Specification (ASE_TSS.1)
Design (ADV)	Basic Functional Specification (ADV_FSP.1)
Guidance (AGD)	Operational User Guidance (AGD_OPE.1)
	Preparative Procedures (AGD_PRE.1)
Lifecycle (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM Coverage (ALC_CMS.1)
	Timely Security Updates (ALC_TSU_EXT.1)
Testing (ATE)	Independent Testing – Conformance (ATE_IND.1)
Vulnerability Assessment (AVA)	Vulnerability Survey (AVA_VAN.1)

Table 2 TOE Security Assurance Requirements

ALC_TSU_EXT.1 Timely Security Updates

Developer action elements:

ALC_TSU_EXT.1.1D
The developer shall provide a description in the TSS of how timely security updates are made to the OS.

ALC_TSU_EXT.1.2D
The developer shall provide a description in the TSS of how users are notified when updates change security properties or the configuration of the product.

Content and presentation elements:

ALC_TSU_EXT.1.1C
The description shall include the process for creating and deploying security updates for the OS software.

ALC_TSU_EXT.1.2C

The description shall include the mechanisms publicly available for reporting security issues pertaining to the OS.

Note: The reporting mechanism could include web sites, email addresses, as well as a means to protect the sensitive nature of the report (e.g., public keys that could be used to encrypt the details of a proof-of-concept exploit).

Evaluator action elements:

ALC_TSU_EXT.1.1E
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3 Security Requirements Rationale

This section provides a rationale that describes how the Security Target reproduced the security functional requirements and security assurance requirements from the protection profile.

EulerOS is a general purpose operating system, and this Security Target is in compliance with the General Purpose Operating Systems Protection Profile, Version 4.1, March 9, 2016 (GP OS PP).

Moreover, as demonstrated in this security target, EulerOS runs on a wide variety of hardware platforms and so it is a general purpose operating system.

5.3.1 Security Functional Requirements Rationale

For SFRs, at first, all the unconditional requirements in the main body of the GP OS PP are coped into this ST; then, as elliptic curves are supported for authentication and key agreement in TLS protocol, the following selection-based requirement is included, which is defined in the annex B in the GP OS PP.

FCS_TLSC_EXT.2 TLS Client Protocol

And because no DTLS is implemented in the TOE, DTLS is not selected in the component FCS_TLSC_EXT.2, so the selection-based requirement

FCS_DTLS_EXT.1 is not necessary to be satisfied, which is not included in this ST.

Because “SSH as conforming to the Extended Package for Secure Shell” is selected for component *FTP_ITC_EXT.1 Trusted channel communication*, the requirements from the *Extended Package for Secure Shell (SSH) Protection Profile, Version 1.0, February 19,2016(EP SSH PP)* are also drawn into this ST, which include the following components based on the selection fact:

FCS_COP.1(SSH) Cryptographic Operation - Encryption/Decryption
FCS_SSH_EXT.1 SSH Protocol
FCS_SSHC_EXT.1 SSH Protocol – Client
FCS_SSHS_EXT.1 SSH Protocol – Server

Finally, no other optional or objective requirements defined in the Annex of the GP OS PP is selected in this ST.

Since all the SFRs are drawn from either the GP OS PP or the EP SSH PP, all dependencies between SFRs are already addressed by the PPs or justified.

Below is the table showing the mapping from protection profile SFRs to security target SFRs.

PP	**PP Requirement**	**ST Requirement**	**Operation & Rationale**
GP OS PP	FAU_GEN.1	FAU_GEN.1	Multiple selections which are allowed by the PP.
	FCS_CKM.1(1)	FCS_CKM.1(1)	Multiple selections which are allowed by the PP.
	FCS_CKM.2(1)	FCS_CKM.2(1)	A selection which is allowed by the PP.

	FCS_CKM.4	FCS_CKM.4	Two selections which are allowed by the PP.
	FCS_COP.1(1)	FCS_COP.1(1)	Multiple selections which are allowed by the PP.
	FCS_COP.1(2)	FCS_COP.1(2)	Multiple selections which are allowed by the PP.
	FCS_COP.1(3)	FCS_COP.1(3)	Multiple selections which are allowed by the PP.
	FCS_COP.1(4)	FCS_COP.1(4)	Multiple selections which are allowed by the PP.
	FCS_RBG_EXT.1	FCS_RBG_EXT.1	Multiple selections which are allowed by the PP.
	FCS_STO_EXT.1	FCS_STO_EXT.1	Copied from the PP without changes
	FCS_TLSC_EXT.1	FCS_TLSC_EXT.1	Multiple selections which are allowed by

			the PP.
	FCS_TLSC_EXT.2	FCS_TLSC_EXT.2	Copied from the PP without changes.
	FDP_ACF_EXT.1	FDP_ACF_EXT.1	Copied from the PP without changes.
	FDP_IFC_EXT.1	FDP_IFC_EXT.1	A selection which is allowed by the PP.
	FMT_MOF_EXT.1	FMT_MOF_EXT.1	Copied from the Technical Decision #0104 without changes.
	FMT_SMF_EXT.1	FMT_SMF_EXT.1	Refinements, selections and assignments which are allowed by the Technical Decision #104.
	FPT_ACF_EXT.1	FPT_ACF_EXT.1	Copied from the PP without changes.
	FPT_ASLR_EXT.1	FPT_ASLR_EXT.1	Copied from the PP without changes.
	FPT_SBOP_EXT.1	FPT_SBOP_EXT.1	Copied from the PP

			without changes.
	FPT_TST_EXT.1	FPT_TST_EXT.1	Multiple selections which are allowed by the PP.
	FPT_TUD_EXT.1	FPT_TUD_EXT.1	Copied from the PP without changes.
	FPT_TUD_EXT.2	FPT_TUD_EXT.2	Copied from the PP without changes.
	FIA_AFL.1	FIA_AFL.1	Two selections and an assignment which is allowed by the PP.
	FIA_UAU.5	FIA_UAU.5	Multiple selections which are allowed by the PP.
	FIA_X509_EXT.1	FIA_X509_EXT.1	A selection which is allowed by the PP.
	FIA_X509_EXT.2	FIA_X509_EXT.2	A selection which is allowed by the PP.
	FTP_ITC_EXT.1	FTP_ITC_EXT.1	Multiple selections and an

			assignment which are allowed by the PP.
	FTP_TRP.1	FTP_TRP.1	Multiple selections which are allowed by the PP.
EP SSH PP	FCS_COP.1(1)	FCS_COP.1(SSH)	Multiple selections and an assignment which are allowed by the PP.
	FCS_SSH_EXT.1	FCS_SSH_EXT.1	Multiple selections and an assignment which are allowed by the PP.
	FCS_SSHC_EXT.1	FCS_SSHC_EXT.1	Multiple selections and an assignment which are allowed by the PP.
	FCS_SSHS_EXT.1	FCS_SSHS_EXT.1	Multiple selections and an assignment which are allowed by the PP.

5.3.2 Security Assurance Requirements Rationale

The statement of security assurance requirements (SARs) found in section *5.2 TOE Security Assurance Requirements*, is in strict conformance with the General Purpose Operating Systems Protection Profile.

5.3.3 Rationale for Conformance to Protection Profile

This Security Target is in compliance with the General Purpose Operating Systems Protection Profile, Version 4.1, March 9, 2016 (GP OS PP).

For all of the content incorporated from the protection profile, the corresponding rationale in that protection profile remains applicable to demonstrate the correspondence between the TOE security functional requirements and TOE security objectives. Moreover, as demonstrated in this security target EulerOS can run on a wide variety of hardware platforms being only one the evaluated, so it is a general purpose operating system.

6 TOE Summary Specification (TSS)

This section describes security functions of EulerOS. Security Functions (SFs) in EulerOS satisfy the security functional requirements of the protection profile. The TOE also includes additional relevant security functions which are also described in the following sections, as well as a mapping to the security functional requirements satisfied by the TOE.

This section presents the TOE Security Functions (TSFs) and a mapping of security functions to Security Functional Requirements (SFRs). The TOE performs the following security functions:

1) Cryptographic Support
2) User Data Protection
3) Security Management
4) Protection of the TSF
5) Audit
6) Identification and Authentication
7) Trusted Path/Channels

6.1 Cryptographic Support

The TOE offers different cryptographic services in kernel, and provides a socket interface to user space applications. In addition, it also provides cryptographic algorithms for general use in user space.

The following subsections cover the different types of cryptographic services analyzed as part of the evaluation. Additional cryptographic mechanisms are active in the TOE, which are however not subject to the assessments of this evaluation.

6.1.1 Cryptographic Algorithms and Operations

1) Random number generation

Deterministic random bit generation (DRBG) is implemented in accordance with NIST Special Publication 800-90A. All three viable DRBGs (HMAC, Hash and CTR) defined in the standard are implemented in EulerOS.

There are several hardware and software entropy pools in EulerOS, considering data from external events, such as timing interrupt requests, disk and network I/O, as well as human input from keyboard and mouse. The main source of entropy in the system is the CPU cycle counter which continuously tracks hardware interrupts. Other hardware-dependent entropy source is the TPM. Each entropy source is independent of the other sources and does not depend on time. Next table summarized the hardware entropy bits coming form the evaluated platform:

Evaluated platform	CPU interrupts (/dev/hwrng)	TPM (/dev/tpm0)
FusionServer RH2288H V3 Rack Server	2044 bits	512 bits

All these entropy sources feed the input pool. The input pool maintains a maximum internal state of 4096 bits. The current state can be checked at every moment at:

- /proc/sys/kernel/random/entropy_available

The entropy data is obtained from the entropy sources in a raw format and is conditioned before using it as input for the DRBG. The entropy data is hashed (sha-1) as part of the mixing functions in the cryptographically secure pseudorandom number generator (CSPRNG). In particular the entropy data (after conditioning, 160bits) together with other data (such as the nonce) seed the DRBG algorithm. Considering that the NIST requires at least 112 bits for the generation of a random number (using and approved algorithm (DRBG)), the DRBG seed of the EulerOS is considered valid. Min-entropy value for the evaluated hardware platform is checked in the testing report (aka Assurance Activity Report).

The DRGB algorithms have been certified (see certificate number in the table 3), i.e., these algorithms meet the SP 800-90A, including the health-tests defined this standard (section 11.3).

The TSF defends against tampering of the random number generation (RNG) / pseudorandom number generation (PRNG) sources by encapsulating its use in Kernel Security Device Driver.

2) Other crypto operations

The encryption and decryption operations are performed by independent kernel modules. Besides, the TSF provides other cryptographic operations such as hashing and digital signatures. Hashing is used by other algorithms implemented in EulerOS (the hashed message authentication code, RSA, DSA, and ECDSA signature services and elliptic curve Diffie-Hellman key agreement, and random bit generation). When EulerOS needs to establish an RSA-based shared secret key it can act both as a sender or recipient, any decryption errors which occur during key establishment are presented to the user at a highly abstracted level, such as a failure to connect.

The table below gives the cryptographic algorithm standards EulerOS supports:

Cryptographic Operation	Standard	NIST Certs.
Encryption/Decryption	FIPS 197 AES (For CBC, KW, XTS, and GCM modes)	#5066
Digital signature	FIPS 186-4 RSA	#2746
Digital signature	FIPS 186-4 ECDSA	#1312
Hashing	FIPS 180-4 SHA-1/SHA-256/SHA-384/SHA-512	#4126
Keyed-Hash Message Authentication Code	FIPS 198-2 HMAC	#3381
Random number generation	NIST SP 800-90 CTR_DRBG/Hash_DRBG/HMAC_DRBG	#1884
ECC Key agreement Scheme	NIST SP 800-56A	#1620
Key Transport Scheme	NIST SP 800-56B	Tested by the CC evaluation lab

Table 3 Cryptographic Algorithm Standards supported by EulerOS

3) Key management

EulerOS kernel provides a Key Retention Service specifically to host and manage secret and private keys to mitigate tampering or access to sensitive key materials for user-mode processes. Each key has a type, a serial number, access control permission, an expiry time and other attributes. Users can add, request, invalidate and revoke a key using the interfaces provide by the service. The service includes a background garbage collector; all dead, revoked and expired keys will be garbage collected after a certain amount of time has passed. When a key of type *trusted* or *encrypted* (which are regarded as *critical*) is garbage collected, the RAM area its payload occupies is overwritten with all zeroes.

Services in user space can maintain some keys on their own, without help of the kernel facilities. DRBG is initialized when the services start to create asymmetric and symmetric keys. The table below gives a summary of the key management of some main services in the TOE:

Service	Keys	Key generation	Key Storage	Key Entry/output	Key Zeroization
EVM	Trusted key	Using SP 800-90A DRBG	Service's memory	API input/output parameters and return values are constrained within the service	Zeroized when freeing the cipher handler or after removing the power supply.
			On disk (sealed by TPM)	N/A	N/A
	Evm key	Using SP 800-90A DRBG	Service 's memory	API input/output parameters and return	Zeroized when freeing the cipher handler after removing the power

				values are constrained within the service	supply
			On disk (after being encrypted by trusted key using AES256-CBC)	N/A	N/A
TLS	Session keys (Symmetric keys) and HMAC keys	Key derivation	Service's memory	API input/output parameters and return values are constrained within the service	Zeroized when session ends (by calling DestroyContext (ctx, freeit) or HMAC_Destroy()) or after removing the power supply
	Private/ public Asymmetric keys	Use SP 800-90A DRBG and RSA/DSA/ECDSA key generation mechanism in the service	Service's memory	API input/output parameters and return values are constrained within the service	Zeroized when freeing the cipher handler or after removing the power supply
			On disk (key3.db, cert8.db)	N/A	N/A
SSH	Session	key derivation	Service's	API	Zeroized when

	keys and HMAC keys		memory	input/output parameters and return values are constrained within the service	session ends or after removing the power supply
	Private/ public Asymmetric keys	Use SP 800-90A DRBG and RSA/ECDSA key generation mechanism in the service	Service's memory	API input/output parameters and return values are constrained within the service	Zeroized when freeing the cipher handler or after removing the power supply
DM-crypt	Master key	SP 800-90A DRBG	Kernel memory	API input/output parameters and return values are constrained within the kernel	Zeroized when the file system is un-mounted or after removing the power supply [Note: the type of the key is ‘trusted’]

Table 4 Origin/storage/zeroization of keys

All the keys maintained by the kernel and applications exist in system RAM and will be removed when the memory lost its power. All keys that exist on non-volatile memory (e.g. disks) are stored after encryption, which depends on the application implementation.

4) Hash functions

The TOE implements 4 hashing functions with different sizes (SHA1, SHA256, SHA384 and SHA512). These functions are implicitly used in other cryptographic operations (see KHMAC and signature creation and versification). All 4 hashes can be used for all the associated operations included in FCS_COP.

6.1.2 Cryptographic network services

The TOE provides cryptographic secured network communication channels, based on the following main libraries:

- OpenSSH: The OpenSSH application suite provides access to the command line interface of the TOE. OpenSSH can provide interactive as well as non-interactive sessions, and the console provided via OpenSSH provides the same environment as a local console. OpenSSH implements the SSHv2 protocol. The cryptographic primitives are provided by OpenSSL;
- Libreswan / Kernel: The Libreswan application suite implements the IKEv1 and IKEv2 protocols to securely establish the symmetric keys used for an IPSEC tunnel. These keys are handed to the kernel which implements the IPSEC protocol.
- OpenSSL: Provides the cryptographic base for the TLS and SSH protocols.

TLS

The TOE implements TLS protocol to enable a trusted network channel that is used for client and server authentication, as well as HTTPS. The following table summarizes the TLS RFCs implemented in EulerOS:

RFC#	Name & Link	Comment
6101	The Secure Sockets Layer (SSL) Protocol Version 3 (SSL3) https://tools.ietf.org/html/rfc6101	Requirements for SSL3; Made obsolete by TLS protocols.
2246	The TLS Protocol Version 1.0 https://tools.ietf.org/html/rfc22	Requirements for TLS 1.0

	46	
3268	AES Cipher suites for TLS https://tools.ietf.org/html/rfc3268	Extension to TLS1.0, adding Advanced Encryption Standard (AES) cipher suites.
3546	Transport Layer Security (TLS) Extensions https://tools.ietf.org/html/rfc3546	Extension to TLS1.0, adding a mechanism for negotiating protocol extensions during session initialization. Made obsolete by RFC 4366.
4366	Transport Layer Security (TLS) Extensions https://tools.ietf.org/html/rfc4366	Extension to TLS1.0 and TLS 1.1, specifying a set of specific extensions and a generic extension mechanism.
4346	The Transport Layer Security (TLS) Protocol Version 1.1 https://tools.ietf.org/html/rfc4346	Requirements for TLS 1.1;
4492	Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) https://tools.ietf.org/html/rfc4492	Extensions to TLS 1.1, specifying the ECC cipher suite.
4681	TLS User Mapping Extension https://tools.ietf.org/html/rfc4681	Extends TLS to include a User Principal Name during the TLS handshake.
5246	The Transport Layer Security (TLS) Protocol Version 1.2 https://tools.ietf.org/html/rfc5246	Requirements for TLS 1.2;
5289	TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode	Extensions to TLS 1.2, adding SHA-256/384 for hashing and GCM mode for authenticated encryption with additional data

	(GCM) https://tools.ietf.org/html/rfc5289	(AEAD) cipher.

Table 5 TLS RFCs implemented in EulerOS

EulerOS implements HTTPS as described in RFC 2818 (https://tools.ietf.org/html/rfc2818) so that applications running on the TOE can securely connect to external servers using HTTPS protocol.

The complete set of TLS cipher suites implemented in EulerOS, which are also used in the evaluated configuration, are as follows:

TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,
TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246.

According to these implemented TLS cipher suites, the following asymmetric cryptographic key generation schemes are used in TLS protocol:

RSA schemes:

The TOE supports the generation of RSA keys for the TLS protocol using the *certutil*(1) application; the RSA keys sizes supported include:

a) 2048 bits (default),

b) 3072 bits.

ECC schemes:

The TOE supports the generation of ECDSA keys for the TLS protocol. The EC key sizes supported are those specified in the curves below:

a) NIST p-256 (256 bits),

b) NIST p-384 (384 bits),

c) NIST p-521 (512 bits).

All the asymmetric cryptographic key generation schemes together with the key sizes supported also apply to the IKE protocol to generate key pairs.

For key exchange methods, the TOE supports ECDHE_RSA, ECDHE_ECDSA, and RSA. If RSA-based key exchange scheme is used in TLS, while the TOE performing the key material decryption, for any errors detected resulting from error input or error decryption, the API would always report a fixed error number *SECFailure* (=-1), and give an ambiguous error information, revealing no particular error that occurred.

Hostname verification before establishing TLS connection is supported in the TOE. The TLS client will check if the DNS name (part of Subject Alternative Name) or the Common Name in the server's certificate matches the specified hostname. IP addresses as a reference identifiers are only supported for LDAP connections. By default, wildcards are supported in the comparing and they match only in the left-most label. If neither can match, the server is regarded as a fraud and the client will then give up the connection; otherwise, the connection can be created as usual. EulerOS does not provide a general-purpose certificate pinning capability.

SSHv2 Protocol

The TOE also implements SSHv2 protocol to enable users from a remote host to establish a secure connection and perform a logon to the TOE. The following table summarizes the SSH RFCs implemented in EulerOS. The requirements of the relevant standards explain the different implementation choices such as optional features.

RFC#	Name & Link	Comment
4253	The Secure Shell (SSH) Transport Layer Protocol https://tools.ietf.org/html/rfc4253	Requirements for SSHv2.
4252	The Secure Shell (SSH) Authentication Protocol https://www.ietf.org/rfc/rfc4252.txt	Requirements for SSH authentication protocol.
4251	The Secure Shell (SSH) Protocol Architecture https://www.ietf.org/rfc/rfc4251.txt	Requirements for SSH Protocol Architecture.
4254	The Secure Shell (SSH) Connection Protocol https://www.ietf.org/rfc/rfc4252.txt	Requirements for SSH Connection Protocol.

According to RFC 4253, the maximum packet size for SSH protocol should be 32768 (32K) bytes or more. The TOE implements SSH protocol using the OpenSSH facility, which sets the maximum packet size to *262144 (256K)* bytes for SSHv2. For each received packet, the packet length will be checked after decryption. If the length of a received packet exceeds 256K bytes, the recipient would terminate.

The TOE supports the following security functions of the SSHv2.0 protocol:

- AS a SSH client or server, the TOE supports two different user authentication methods, i.e., public key-based and password-based.
- The TOE implements the SSHv2 protocol with the following algorithms:
    - Encryption algorithms: only aes128-ctr, aes256-ctr, aes128-cbc, aes256-cbc, AEAD_AES_128_GCM, AEAD_AES_256_GCM are used. For aes128-ctr and aes256-ctr, a counter with length AES_BLOCK_SIZE (=16) bytes is setup. For each cipher operation,

the counter is incremented by one, overflow ignored. This can make the counter unique.

- o MAC algorithms: only hmac-sha1, hmac-sha2-256, hmac-sha2-512 are used.
- o Public key algorithms for authentication of host and user: ssh-rsa, ecdsa-sha2- nistp256, ecdsa-sha2-nistp384. The key sizes for these algorithms used are as follows:
  a) Key sizes for RSA: 2048 bits, 3072 bits;
  b) Key sizes for ECDSA and the corresponding curves: NIST p-256 (256 bits), NIST p-384 (384 bits);
- o Key exchange methods: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521.

6.1.3 Data Protection

EulerOS provides data protection APIs in *openssl* package, which applications can use to protect any persisted data that the developer deems to be sensitive. Below are some of the *openssl* APIs available:

- EVP_EncryptInit, EVP_EncryptUpdate, EVP_EncryptFinal;
- EVP_DecryptInit, EVP_DecryptUpdate, EVP_DecryptFinal;
- EVP_EncryptInit_ex, EVP_EncryptUpdate_ex, EVP_EncryptFinal_ex;
- EVP_DecryptInit_ex, EVP_DecryptUpdate_ex, EVP_DecryptFinal_ex;

The AES CBC encryption algorithm is used by default in the APIs above.

Additional information about EVP API, can be found here.

For protection of data on disks, EulerOS offers an additional layer between the file systems and the physical block device, which is used to encrypt and decrypt any data transmitted between the file system and the block device. This is done by functionality *dm_crypt* (LUKS extension) using *device mapper*.

Before mounting an encrypted block device, the owner has to provide a passphrase. This passphrase is used to decrypt the symmetric **master volume key** which is injected into the kernel. Using that master volume key, the kernel can decrypt (to unlock) the block device and provides access to data stored on that block device. At this point, the encrypted block device can be mounted as usual, and written data to the device is encrypted and read data from the device is decrypted transparently by the kernel using the master volume key. When the encrypted block device is un-mounted and locked (i.e. the kernel is informed to discard the master volume key), no user, including administrative users like the root user, is able to access any data on it any more. When an administrator would access the raw hardware hosting the block device, only encrypted data can be read.

For the cryptographic operations for data on encrypted devices, the creator of the encrypted block device can select the cipher. The master volume key is obtained from a random number generator and stored on the block device encrypted with the user's passphrase. A tool, *cryptsetup(8)*, can be used to erase the storage location of an encrypted master volume key, which implies that the user owning the passphrase of the affected encrypted session key is not able to unlock the block device any more.

Besides LUKS-encrypted block device, some other persistent data are regarded sensitive and then encrypted. All of them are listed in the table below:

Data	Usage	Protection	Storage
LUKS master volume key	protect an encrypted block device	Encrypted by default with aes-xts with key size 256.	Stored in a key slot of the LUKS device
User certificate for SSH	User authentication	Encrypted with AES-128-CBC	File *id_rsa* or *id_dsa* in user’s home directory
EVM key (or EVM user key)	Protect extended attribute of a file in data integrity verification	Encrypted by the EVM trusted key using AES256-CBC	In file */root/evm-key*

6.1.4 SFR Summary

- FCS_CKM.1(1), FCS_CKM.2(1), FCS_COP.1(1), FCS_COP.1(SSH), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1: the Cryptographic Algorithm and its Standards used in EulerOS.
- FCS_CKM.4: EulerOS overwrites critical cryptographic keys at garbage collection time.

- FCS_STO_EXT.1: EulerOS provides crypto APIs (eg, *openssl* library) for developers and the *cryptsetup(8)* tool for system administrators to encrypt and decrypt sensitive data.
- FCS_TLSC_EXT.1, FCS_TLSC_EXT.2: EulerOS implements TLS to provide server and mutual authentication, confidentiality and integrity to upper-layer protocols such as Extensible Authentication Protocol and HTTPS.
- FCS_SSH_EXT.1, FCS_SSHC_EXT.1, FCS_SSHS_EXT.1: EulerOS implements SSHv2 to provide security network communication channel.

6.2 User data protection

6.2.1 Discretionary Access Control

The general security policy determines that subjects (i.e., processes) are allowed only the accesses specified by the policies applicable to the object the subject requests access to. Further, the ability to propagate access permissions is limited to those subjects who have that permission, as determined by the policies applicable to the object the subject requests access to.

A subject may possess one or more of the following capabilities which provide the following exemptions from the DAC mechanism:

- CAP_DAC_OVERRIDE: A process with this capability is exempt from all restrictions of the discretionary access control and can perform any action desired. For the execution of a file, the permission bit vector of that file must contain at least one execute bit.
- CAP_DAC_READ_SEARCH: A process with this capability overrides all DAC restrictions regarding read and search on files and directories.
- CAP_CHOWN: A process with this capability is allowed to make arbitrary changes to a file's UID or GID.
- CAP_CHOWN: Setting permissions and ownership on objects even if the process' UID does not match the UID of the object.
- CAP_FSETID: Don't clear SUID and SGID permission bits when a file is modified.

DAC provides the mechanism that allows users to specify and control access to objects that they own. DAC attributes are assigned to objects at creation time and remain in effect until the object is destroyed or the object attributes are changed. DAC attributes exist for all types of named object known to the TOE. DAC is implemented with permission bits.

The DAC mechanism applies only to named objects which can be used to store or transmit user data. Other named objects are also covered by the DAC mechanism but may be supplemented by further restrictions. These additional restrictions are out of scope for this evaluation. Examples of objects which are accessible to users that cannot be used to store or transmit user data include:

- Virtual file systems, which give user space processes an interface to kernel data structures (such as most of *procfs*, *sysfs*, *binfmt_misc*);
- Process signals;

During creation of objects, the TSF ensures that all residual content is removed from that object before making it accessible to the subject requesting the creation.

When data is imported into the TOE (such as when mounting disks created by other trusted systems), the TOE enforces the permission bits applied to the file system objects.

6.2.1.1 Permission bits

The TOE supports standard UNIX permission bits to enforce one form of access control for file system objects in all supported file systems. For each process, it is bound with one user (denoted as process *uid*) and several groups (denoted as several *gid*s), indicating the owner of the process and the groups the process belongs to. For each file object, there are three categories of users and three types of access. The three user categories are the owning user (u), the owning group (g), and other users (o); the three access types are read (r), write (w) and execute (x).

Each file object has in its metadata a uid for the owning user, a gid for the owning group, and a 9-bit permission vector. The permission vector contains three 3-bit subvectors, defining the allowed access types to the object for the owning user, owning group and other users separately. Each bit in a subvector corresponds to one of the access types; if a bit in the subvector is

1, it means the object is allowed to be accessed with the corresponding access type.

The three user categories are given different priorities, which are the owning user, the owning group, and others in descending order. For access decision, only the subvector for the user category with the highest priority is checked against the requested access type. The access decision is made in kernel space when a process issues an access request to a file object. The decision is made as follows:

1) Decide the user category of the process to the file object:
   i. If the process uid equals to the owning user of the file object, the user category is *owning user*; otherwise
   ii. For each of the groups the process belongs to, if the gid equals to the owning group of the file object, the user category is *owning group*; otherwise
   iii. The user category is *other users*;
2) Get the subvector in file metadata for the user category;
3) If the bit for the requested access in this subvector is 1, then grant this access; otherwise deny it;

Besides, there are several special cases for the write access:

- Write access to file systems mounted as read only (e. g. CD-ROM) is always rejected (the exceptions are character and block device files, which can still be written to, however the write operations do not modify the information on the storage media);
- Write access to file system objects marked as immutable is always rejected.

Besides the 9-bit permission vector, a SAVETXT bit (or sticky bit) is used only for world-writable temp directories, preventing the removal of files by users (and the system administrator) other than the owner in the directories.

Each process has an inheritable “umask” attribute which is used to determine the default access permissions for new objects it creates. It is a bit mask of the 9-bit permission vector, and specifies the access bits to be removed from the permission vector for new objects. For example, setting the umask to “002” ensures that new objects will be writable by the owner and group, but not by others. The umask is defined by the administrator in the /etc/login.defs file, or default to 022 if it is not specified explicitly.

6.2.1.2 Access check for file system objects

Access to file system objects is generally governed by the 9-bit permission vector. Access to a file system object is checked when the object is initially opened, and is no longer checked for each subsequent access. Changes to access controls (e.g., revocation) begin to be effective only with the next attempt to open the object.

6.2.1.3 Access check for IPC objects

The TOE implements the following standard types of IPC mechanisms:

- SYSV Shared Memory
- SYSV and POSIX Message Queues
- SYSV Semaphores

Each of the IPC objects is also bound with a 9-bit permission vector, which controls access to the IPC object.

There are other IPC objects in the TOE, including UNIX domain socket and Named Pipes, which are represented as file system objects. The access control mechanism covering file system objects are also applicable to these IPC mechanisms.

6.2.1.4 Access check for at and cron jobs queues

at and *cron* jobs can only be accessed (read/added/modified/deleted) by the owning user. The TOE maintains *at* and *cron* job queues for each user. The *at* or *cron* jobs are started with the UIDs/GIDs of the creator of the job.

The root user can always access every *at* or *cron* job queue.

6.2.2 VPN client

EulerOS does not provide a VPN client; however, it provides related APIs in package *libreswan*, which can be used to enable VPN clients to protect IP traffic using the IPsec tunneling protocol. Iptables is a packet filter implemented as part of the network stack in the OS kernel. The user space application *iptables(1)* allows the configuration of the IPTables kernel components, and then can be used to configure all IP traffic that is routed through the IPsec tunnel except for:

- IKE traffic used to establish the VPN tunnel;
- IPv4 ARP traffic for resolution of local network layer addresses and to establish a local address;
- IPv6 NDP traffic for resolution of local network layer addresses and to establish a local address;

Libreswan implements IPsec in user space. It implements Internet key exchange (IKE) protocol version 1 and 2 in a user-level daemon. Libreswan interfaces with the EulerOS kernel using netlink to transfer the encryption keys. Packet encryption and decryption is done in the kernel.

6.2.3 SFR Summary

- FDP_ACF_EXT.1: EulerOS provides a Discretionary Access Control policy to limit reading, modification, and execution of objects by non-authorized users.
- FDP_IFC_EXT.1: EulerOS provides iptables APIs in package libreswan, allowing VPN clients to protect IP traffic on the VPN tunnel using IPsec protocol.

6.3 Security Management

The following table lists which activities can be done by a EulerOS user or a local administrator. A checkmark indicates which entity can invoke the management function. General users, or programs running on their behalf, are not able to modify policy or configuration that is set by the privileged administrator, which results in that the user cannot override the configuration specified by the administrator.

Management Function	**FMT_SMF_EXT.1**	**FMT_MOF_EXT.1**

Enable/disable screen lock	M	O
Configure screen lock inactivity timeout	M	O
Configure local audit storage capacity	M	X
Configure minimum password Length	O	X
Configure minimum number of special characters in password	O	X
Configure minimum number of numeric characters in password	O	X
Configure minimum number of uppercase characters in password	O	X
Configure minimum number of lowercase characters in password	O	X
Configure remote connection inactivity timeout	O	X
~~enable/disable unauthenticated logon~~	~~O~~	~~M~~
Configure lockout policy for unsuccessful authentication attempts through [**selection:**	O	X

timeouts between attempts, limiting number of attempts during a time period]		
Configure host-based firewall	O	X
Configure name/address of audit/logging server to which to send audit/logging records	O	X
Configure audit rules	O	X
Configure name/address of network time server	O	O
~~Enable/disable automatic software update~~	~~O~~	~~O~~
~~Configure WiFi interface~~	~~O~~	~~O~~
~~Enable/disable Bluetooth interface~~	~~O~~	~~O~~
Configure USB interfaces	O	X
~~Enable/disable [**assignment**: list of other external interfaces*]*~~	~~O~~	~~O~~
[**assignment**:*none]*	O	O

6.3.1 SFR Summary

- FMT_MOF_EXT.1: EulerOS provides the user with the capability to administer the security functions described in the security target. The mappings to specific functions are described in each applicable section of the TOE Summary Specification.

6.4 Protection of the TSF

6.4.1 Separation and Domain Isolation

The TSF provides a security domain for its own protection and provides process isolation. The security domains used within and by the TSF consists of the following components:

- Hardware
- Kernel-mode software
- Trusted user-mode processes
- User-mode Administrative tools process

The TSF hardware is managed by the TSF kernel-mode software and is not modifiable by untrusted subjects. The TSF kernel-mode software is protected (from read and write) by hardware execution state. The TSF hardware provides a software interrupt instruction that causes a state change from user mode to kernel mode. The TSF kernel-mode software is responsible for processing all interrupts, and determines whether or not a valid kernel-mode call is being made. In addition, the memory protection mechanism enforced by the hardware ensures that attempts to access kernel-mode memory from user mode would result in a hardware exception, which would be handled by kernel-mode software. In this way, the kernel can ensure that kernel-mode memory cannot be directly accessed by software not executing in the kernel mode. This results in separation of kernel space and user space.

The TSF provides process isolation for all user-mode processes through private virtual address spaces (private process page tables), execution context (registers, program counters, resource usage registrations), and security context (process credentials, capabilities, control group information, security attributes). The data structures defining process address space, execution context and security context are all stored in protected kernel-mode memory. All security relevant privileges are considered to enforce TSF Protection. All user-mode processes share the kernel space (although it is unable to access kernel space if not permitted by the kernel), and each user-mode process has its own private memory space. This is the isolation of memory spaces of different user-mode processes. So, user-mode processes are isolated from each other.

User-mode administrator tools execute with the security context of the process running on behalf of the authorized administrator. Administrator processes are also protected like other user-mode processes, by process isolation.

The TSF implements cryptographic mechanisms within kernel space; and the services can be accessed by both kernel- and user-mode components in order to isolate those functions from the rest of the TSF to limit exposure to possible errors while protecting those functions from potential tampering attempts.

6.4.2 Protection of OS Binaries, Audit and Configuration Data

All the files in below directories are owned by the system administrator and the permission bits are set in a way that only the system administrator is allowed to modify them:

Kernel images: /boot/
Kernel drivers: /lib/modules/`uname –r`/kernel/
Security audit logs: /var/log/audit/
Shared libraries: /lib, lib64
System executables: /usr/bin, /usr/sbin
System configuration files: /etc/

6.4.3 Protection from Implementation Weaknesses

The user-mode processes in EulerOS implement Address Space Layout Randomization (ASLR) mechanism in order to load executable code at unpredictable base addresses. The base address is generated using a pseudo-random number generator that is seeded by high quality entropy sources when available which provides 8 or 28 random bits (for 32-bit and 64-bit architecture separately) for memory mapping. The binaries are compiled with *-pie -fPIE* option to use ASLR feature.

A stack buffer overrun protection capability (CC_STACKPROTECTOR) is built in the kernel that will terminate a process after a potential buffer overrun on the thread’s stack is detected by a failed verification of the canary data in the stack frame of each function call. All EulerOS executable binaries developed in C/C++ implement this protection by being complied with the *-fstack-protector-strong* option.

Note: If a symbol *_stack_chk_fail* is found in an ELF executable, this indicates that the executable is built with *-fstack-protector-strong* option. However, this does not hold vice versa. The compiler (GCC) can disable this option for some invocations if *–O* option is enabled and it deems that there is no possibility to overflow the stack (e.g. if there is no array on the stack). In addition to that, there are other situations where SBOP protections are not applied. They have been classified in the following types:

- Type 1: There are not functions in the file at all.
- Type 2: The ELF file does NOT use character arrays on stack.
- Type 3: The stack protection mechanism is disabled by an explicit attribute 'always_inline’.
- Type 4: The ELF file uses character arrays on stack, but also uses secure functions to operate on the array.
- Type 5: The ELF file is for a special programming language, which has special mechanisms to avoid stack smashing.

In Appendix A there is a list of all the files affected as well as the rationale for each of them that can be one of the previous types.

6.4.4 Platform Integrity and Code Integrity

Integrity of kernel image, kernel modules and system executables is verified using the **Secure Boot** mechanism. No special hardware is needed for Secure Boot.

The trust chain at boot time looks like this:

CRTM -> UEFI -> shim -> GRUB2 -> OS kernel

The first two parts, CRTM and UEFI, are for hardware and firmware, and are out of the scope of this ST. One or more platform certificates (called PKs) are integrated into firmware, which are the root of the trust chain above. There is the following booting stage which covers the trust chain for system integrity check at boot time.

1. Secure Boot stage for booting integrity

- Pre-booting. *Shim* is the first-stage of boot loader, containing a self-signed CA certificate, which is generated with RSA and key size 2048, and used to verify GRUB2 and OS kernel image. It is signed by one of the PKs. At this stage, *shim* is loaded and verified by the firmware using one PK. If the verification fails, an error message will be prompted and the boot process gets blocked; otherwise, *shim* will be launched, and try to load GRUB2. Actually this pre-booting is out of the scope of this ST.
- Booting stage 2. Once verified and run by firmware, *shim* tries to load GRUB2. After loading, *shim* begins to verify GRUB2 using the self-signed CA embedded in *shim*. If the verification of GRUB2 failed, an error will be reported and the boot process will terminate; otherwise, *shim* will launch GRUB2 and the boot process continues.
- Booting stage 3. If launched, GRUB2 will load OS kernel and try to verify it. Generally, OS kernel is signed by the vendor using the same key as the one for signing GRUB2. GRUB2 calls back into *shim* to verify the signature of the kernel. If the signature cannot be verified, an error will be shown, and the booting process will be blocked; otherwise, the kernel will be started.
- Loading kernel modules. There is another certificate integrated in OS kernel, which is generated with RSA and key size 2048. All kernel modules are signed by the OS vendor using the private key paired with the certificate, and will always be verified using the certificate before being loaded into kernel. Furthermore, the TSF includes a Code Integrity Verification mechanism, whereby kernel modules will be loaded only if they are digitally signed by either Huawei or a trusted root certificate authority recognized by Huawei. This mechanism uses public-key cryptography technology to verify the digital signature of each kernel module when it is to be loaded.

  When trying to load a kernel module, the TSF compares two hash values of the module: one is the decrypted version of the value included within the module file using the public key stored in the certificate; the other is computed based on the code in the kernel module using cryptographic libraries in the TSF. If the two hash values match, the kernel module can then be loaded as usual. If all modules required are verified successfully by kernel at boot time, the secure boot process can terminate normally, and the first user process can start then. If the verification fails, the kernel will report the error message and will stop the booting process.

It is worth noting that module verification is always done when loading a kernel module, both at boot time and at normal runtime of the system. If a module fails in verification at normal runtime, the load command would abort with an error message, and the kernel would NOT be impacted.

The table below gives a list of protected files for secure boot:

Module / Application	**Protected Files**	**Protected by**	**Configurable by System Manager**
shim	/boot/efi/EFI/BOOT/: BOOTX64.EFI, fallback.efi /boot/efi/EFI/euleros/: MokManager.efi, shim.efi	UEFI (certificate in DB)	No
GRUB2	/boot/efi/EFI/euleros/: grubx64.efi, gcdx64.efi /boot/grub2/**	Shim (vendor certificate)	No
Kernel	/boot/vmlinuz*	shim (vendor certificate)	No
Kernel modules	/usr/lib/modules/3.10.0-xxx/kernel/*.ko	Kernel (vendor certificate)	No

6.4.5 OS and Application Updates

Updates to EulerOS and its applications are delivered as RPM Package Manager files (.rpm files) from the EulerOS repo. These files has a digital signature which can be verified by and end-user to check the identity of the file's publisher.

1. Signature verification in RPM packages

EulerOS uses the tool *rpm(8)* to manage all packages, including the kernel images. Rpm has several modes of operation, including *build*, *query*, *verify*, *install*, *upgrade*, and *erase*.

A RPM package file is in a binary format, and its key parts include:

- A *signature* section, which may contain a GPG signature that can be used to verify that the RPM file has not been modified since it was created.
- A *header* section, containing a set of index entries, which map numbered types to values. This is used for storing internal information about RPM package files.
- A gzip compressed CPIO *archive*, containing the actual files to be installed to the file system.

The GPG signature in the *signature* section of an RPM file is calculated for data found in the *header* and the compressed CPIO *archive* parts when building the RPM package. The private key used by the TOE for signing is generated with RSA algorithm and key size 4096. The related GPG public key is put to a separate rpm package (*gpg-pubkey-381d7ac3-584515cf*), which is exported to the TOE.

A repository is a collection of rpm packages for EulerOS and each EulerOS installation can access one or more repositories remotely or locally using a tool *yum(8)* to get rpm packages for installation or update. A repository is also called a source of *yum*. Before a package is released to a *yum* repository, it is signed by the maintainer (i.e., Huawei for EulerOS). The metadata file of a repository (*repodata/repomd.xml*) is also signed.

There is a configuration file for each yum repository in a EulerOS installation, in which an option for the yum repository can be enabled, *gpgcheck=1*, meaning that GPG verification should be checked for each

RPM file from this repository at installation and update time. To use this option, the public GPG key should be imported into the system (as stated above) and specified in the configuration file. At update time, *yum* will check if the related public GPG key is valid and has not been revoked, and each rpm file downloaded from the repository is then verified using the public GPG key to establish its authenticity.

Updates to EulerOS system and EulerOS applications are delivered through EulerOS repository, which has a collection of software (OS and applications) stored in a Huawei's server (maintainer). EulerOS repository is enabled by default and it is the only distribution channel available; but the user can also modify some of the repository options like changing its URL, enabling/disabling the signature verification or a particular repository among other options. Additionally, the user can go the the repository URL (https://devcenter.huawei.com/ict/site-euleros/euleros/repo/yum/2.0/os/x86_64) to search and download EulerOS RPM packages on their own will.

Local RPM package files downloaded by the user can be checked to verify that its signature is valid. The following command can be used by the user to verify a local RPM package: *rpm -K <package name>.* This functionality is performed by default while installing a remote or local RPM package if Yum's configuration file has set the options *gpgcheck=1 and localpkg_gpgcheck=1.*

2. Check certificate ExtendedKeyUsage

Besides the signature protection in RPM, the extended key usage of the certificates in kernel, *codesigning*, is also checked when loading kernel modules. If the built-in certificate contains the *codesigning* purpose and the public key in it matches the signature in the module, the module can then be allowed to insert into kernel.

3. For security update

Huawei Product Security Incident Response Team (PSIRT) manages the receipt, investigation, internal coordination and disclosure of security vulnerability information related to Huawei offerings and it is the only window to disclose the vulnerability of Huawei products.

Vulnerability reporters can submit potential vulnerabilities to Huawei PSIRT by email, and the email addresses are psirt@huawei.com and euler-security@huawei.com.

At same time, for all software in the TOE, both these developed by Huawei and all third-party components, a developer community is created to get feedback from all developers. Developers can report software bugs/vulnerability through a specified web url (https://devcenter.huawei.com/usercenter/addDefectTicket?belong=E236S00P1064T00).

Information in the email is recommended to be encrypted with Huawei's public key (key ID 0x7E3C7F69; PGP fingerprint: DC9D 0676 B289 30A1 D12C 765A 6461 9D10 7E3C 7F69):

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

Version: GnuPG v1.4.7 (MingW32)

mQGiBFAbjoIRBADflj1sw9Dacs6y+r/k6Mg6K84LPpYILV4PdxB/yrOk7r2BfOBJmDp7h9Do90qgyGdH/dXCd3L/CMjgw
2Ec+qOqcqpWrMOyIGWrf9li1pNaRQRmgu0TPw7BoYf0HxjoBisIwb7keZjeww+UFIFZrwuuJn6C1T3UYmBT/3DNawV
Y1wCg351RXB1vceSEuo+0bCXFY7Gd/n8EAKGw0IlBdJ7iblr1UzCq3/FxU5oW4H7T+e305pyoETzo11poqnDMyvYx/8
Bl5+Kmr5UQ4nNn91NFSBAkvUThhoeeSduw582N+80Fe/IAiAoR1cG3fiX+6k0vlRuhsNAYF/Kckygx+gWwPCacrbFz
weMuwh8dBbS4aled9OdzSd4gA/0cIjIVkeW/fRBpj7nSXC3Ia5H802YdKzzNYOTjCUcXQSEKXKiWFnp99En/aXqQm1YT
k/VcUCZEZfZm7FocP/MCWCuQWqM41QfdYf7lRVMPKAWZmDBjDC0x5bfu3Emwp8Q5HXXhMES0FxuOIHITkFFtrRVo
dmIoTCOquylJX4M7gbQYUFNJUlQgPFBTSVJUQGh1YXdlaS5jb20+iGAEExECACAFAlAbjoICGwMGCwkIBwMCBBUCCA
MEFgIDAQIeAQIXgAAKCRBkYZ0Qfjx/aXubAJsHzNg5s8jSD6oIHSkKx13BRGVy9wCg2cdPjcFmSb8GmOn5SyGY/ZwC
MXG5AQ0EUBuOggEIAMhXwVDOBGwuTePBNf/i18mJMOJB+5NVp+K2WizuvJfBxvOugsngQuxlUZP/Sto83iH65ofSyp
MnsIa3IK0XzTiuYkFi+Q1vLZJaTEbf6jkGOmG6z5QeEFxpGCG3xEZam/IISlIN0U6tIu4mG1YIoGdMIyKY0FRq6Ihv4QXR
J2k8D2Iz5mGK9VHVbyhAdUZsdkL31xlzyl508/sMWveqRaeYnerytvY8oNifgfvSkIjtwUkuftVK4oMR2hPNR2MWDq5aN
8ycnb+Vw+zZMz4xjPjFiOjcxwDhOkndjnCvIj4RFE7YsMy8ThNkwItDETr9tTjOEQAERRhVMB+RpPcp5gcAEQEAAYkBa
AQYEQIACQUCUBuOggIbLgEpCRBkYZ0Qfjx/acBdIAQZAQIABgUCUBuOggAKCRCu4my3gtOIaYNdCACF40rx4TAu4n
ZcXPUBnTqBb0GJXAhWkvNDhgRyILo4Rz1RqnDUIHU+fj6B7eQDcWKswSbNWYsVUlXdW+/vRFwF0huqgPaVQMtvFx
rz1ThKrhdypH/ZJon8dGc1JVmGOmbcnc9ZTbcsNwAC+ViR7SffMsI1sJ3Ki9sLo0W4Uep/+xUDtECYq29GDj2IHuWZUJ
Lcxgbbusj/sloDXCdgn38kgJ06dsm9oEqieimIF9n0GaZE20e/R5/NqMYDtAraIuyiKOXkNXyO6LwiuSuiZo0nYv8D6KT2
MUgaym1EA7Mj8IgpQGHmxk9FQw+2BPbGW7n2917P938N6J0vAdjxaKWAzoAAoNYDJaMuTkmDG3+Og6C2cjJhIkT
BAJ0UhEGMa7U0Y7pPHhGqfJgAFo06+g===xaXR

-----END PGP PUBLIC KEY BLOCK-----
```

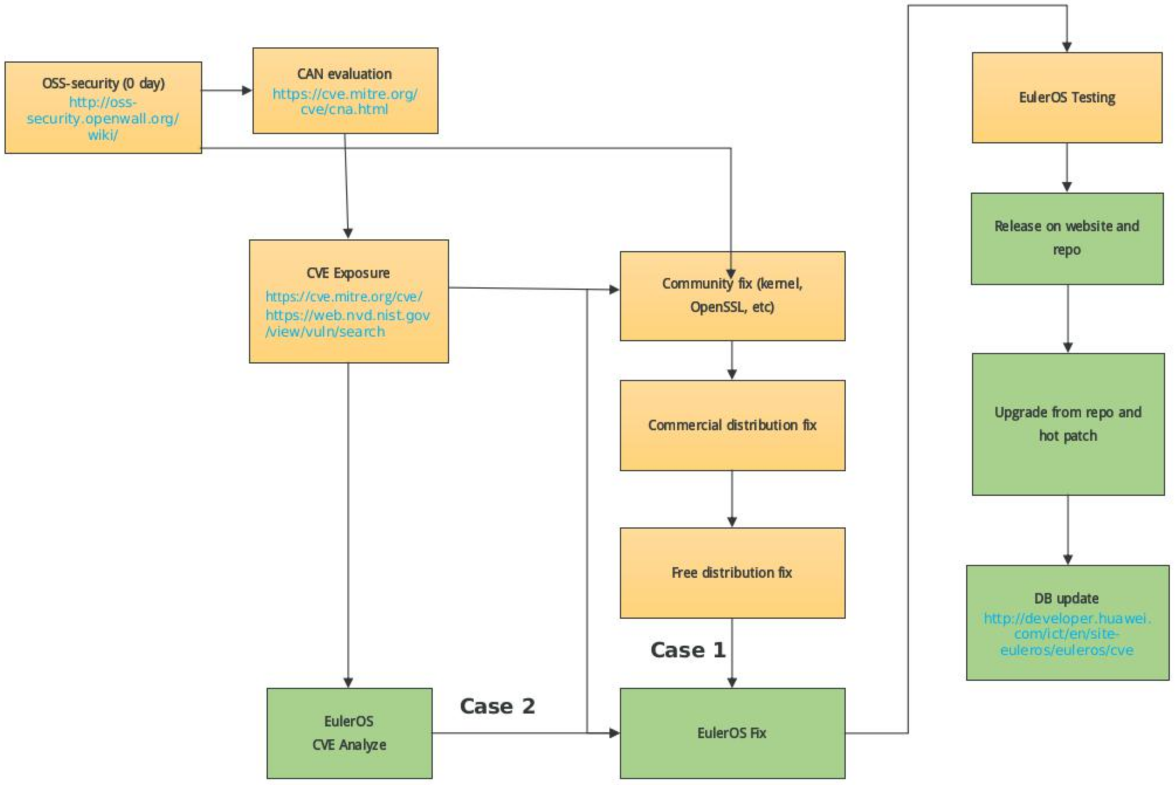
The web url for submitting bugs/vulnerability uses secure communication channel. The bug/vulnerability submitted will not be open to public until the patch or update is open to public.

Besides the vulnerabilities collected from the reporters, PSIRT weekly collect the vulnerability information mainly for third-party components from various sources, such as open source communities, well-known vulnerability library like CVE, etc.

Once PSIRT gets the vulnerability related to the TOE, together with the EulerOS department, they will analyze the severity of the vulnerability, and the severity is classify into 1 through 5 levels. Next, Huawei will develop a patch or an update according to the company level specification “SLA of reaction for vulnerability” which is defined according to the severity level of the vulnerabilities. If there is a patch or update available from the community, then Huawei will adopt it other than developing the patch or update by itself.

Whether the patch or update is self-developed or adopted from the community, Huawei will test it before formal release. According to our SLA, usually the window is about 10 days for the TOE to release a patch or update, while only a few days for severe vulnerability.

The whole process from creating to deploying an update is like this:


OSS-security (0 day)
http://oss-security.openwall.org/wiki/
CAN evaluation
https://cve.mitre.org/cve/cna.html
CVE Exposure
https://cve.mitre.org/cve/
https://web.nvd.nist.gov/view/vuln/search
Community fix (kernel, OpenSSL, etc)
Commercial distribution fix
Free distribution fix
Case 1
Case 2
EulerOS CVE Analyze
EulerOS Fix
EulerOS Testing
Release on website and repo
Upgrade from repo and hot patch
DB update
http://developer.huawei.com/ict/en/site-euleros/euleros/cve

6.4.6 SFR summary

- FPT_ACF_EXT.1: EulerOS provides a Discretionary Access Control policy to limit modification and reading of objects by non-authorized users.
- FPT_ASLR_EXT.1: EulerOS randomizes user-mode process address spaces to hinder address prediction.
- FPT_SBOP_EXT.1: EulerOS executables are compiled with stack overflow protection (compiled with the *-fstack-protector-all* option for native applications).
- FPT_TST_EXT.1: EulerOS checks the integrity of the boot loader, OS loader, kernel image, system binaries, and specified application executable code.
- FPT_TUD_EXT.1, FPT_TUD_EXT.2: EulerOS provides a means to identify the current version of the kernel, the hardware model, and installed applications. The update mechanisms of EulerOS can deliver updated OS kernel and application binaries and let the user confirm that the digital signatures of updated OS kernel and applications are valid.

6.5 Audit

The Lightweight Audit Framework (LAF) is used in the audit subsystem of EulerOS, which is compliant with the requirements from Common Criteria. The EulerOS kernel implements the core of the LAF functionality. It gathers all audit events, analyzes these events based on the audit rules, collects related information, and forwards the audit events that are requested to be audited to the audit daemon executing in user space.

The audit functionality of the Linux kernel is controlled by an audit management tool in user space, which communicates with the kernel through a specific *netlink* channel. This *netlink* channel is usable only by applications with the following capabilities:

- CAP_AUDIT_CONTROL: Performing management operations like adding or deleting audit rules, setting or getting auditing parameters;
- CAP_AUDIT_WRITE: Submitting audit records to the kernel which in turn forwards the audit records to the audit daemon.

The TOE Audit security functionality includes:

Audit event selection
Audit trail
Audit log overflow protection
Audit log access protection

6.5.1 Audit event selection

LAF is able to intercept all system calls and retrieve audit log entries from privileged user space applications. The audit subsystem allows selecting the events to be actually audited from the set of all possibly auditable events based on a group of audit rules. These audit rules are set in an audit configuration file (*/etc/audit/audit.rules*), and will be sent to kernel to control what to be audited.

An audit management tool (*auditctl*(8)) is used to load the audit rules from the audit configuration file and send them to kernel. Generally it is invoked once at boot time. The audit rules in kernel can also be modified at runtime.

6.5.2 Audit trail

An audit record consists of one or more lines of text in a format like “keyword=value”. The following information is contained in all audit record lines:

- Type: indicates the type of the event, such as SYSCALL, PATH, USER_LOGIN, or USER_MGMT;
- Timestamp: Date and time when the audit record was generated;
- Audit ID: unique numerical event identifier;
- Login ID (“auid”): the user ID of the user authenticated by the system (regardless if the user has changed his real and / or effective user ID afterwards);
- Effective user and group ID: the effective user and group ID of the process at the time the audit event was generated;
- Event outcom: Success or failure (where appropriate);
- Process ID of the subject that caused the event (PID);
- Hostname or terminal the subject used for performing the operation;
- Information about the intended operation;

The information above is followed by event specific data. For example, for SYSCALL event records that involve file system objects, multiple text lines will be generated for a single event, all having the same time stamp and audit ID to permit easy correlation.

The audit trail is stored in ASCII text. The TOE provides tools for managing audit trails, which can be used for post-processing of audit data. The tool, *ausearch(8)*, allows selective extraction of records from the audit trail using defined selection criteria. It supports the specification of a fine-grained search pattern where each information component can be searched for, including combinations of these patterns.

6.5.3 Audit log access protection

Access to audit data (including the configuration files and audit trails) by normal users is prohibited by the discretionary access control function of the TOE. The permission is granted only to the system administrator.

6.5.4 SFR summary

- FAU_GEN.1: The TOE audit collection is capable of generating audit events according to configured audit rules. For each audit event, the fields recorded are event type, event ID, the date and time, process ID, user Identifier (auid), event result and other information.

6.6 Identification and Authentication

Each user trying to access a EulerOS instance must have an account on the system. An account is identified by a unique user name, and the name will not be re-used during the whole life cycle of the system. To authenticate a user account, a password is set for it and is saved by EulerOS after encryption.

All logons are treated essentially in the same manner regardless of their forms (e.g., log in remotely using the SSH protocol, log in at the local console) and start with an account name and credentials that must be provided to the TSF. EulerOS authenticates user accounts based on username and password.

For a remote login through the OpenSSH server, the administrator is allowed to enable SSH key-based authentication in addition or instead of the username/password based authentication. When a user can successfully authenticate using the SSH key-based authentication based on a private SSH key in his possession, the TOE grants the user access.

Password-based authentication to EulerOS succeeds when the credential provided by the user matches the stored protected representation of the password.Password authentication can be used for interactive logons and to initiate the “change password” screen.

When the authentication succeeds, the user will be logged onto their desktop, their screen unlocked, or their authentication factors changed depending whether the user logged onto the computer, the display was locked, or the password was to be changed.

6.6.1 PAM-based identification and authentication mechanisms

EulerOS uses a collection of libraries called the "Pluggable Authentication Modules" (PAM) that allow the system administrator to choose how PAM-aware applications authenticate users. The TOE provides PAM modules that implement all the security functionality to:

- Provide login control and set up login ID, UIDs and GIDs for a subject;
- Ensure the quality of passwords;
- Enforce limits for accounts (such as the number of maximum concurrent sessions allowed for a user);
- Enforce the change of passwords after a configured time including the password quality enforcement;
- Enforcement of locking accounts after several successive failed login attempts;
- Restriction of the use of the root account to some specified terminals;
- Restriction of the use of the *su* and *sudo* commands;

It is up to the person or the client application for a remote login to protect the account’s password safely.

When the login process starts, a banner is shown to the user. If the authentication is successful, a session starts for the user; otherwise, some obscured feedback is given. Both failed and successful authentications are logged and audited.

EulerOS maintains a count for the consecutive failed logon attempts. When the count gets larger than a number specified in a PAM policy file, which is 3 by default, EulerOS will lock out the tried account for 300 seconds. EulerOS maintains the number of consecutive failed logon attempts persistently for the account, and hence the number cannot be reset by rebooting the computer.

When the user is successfully authenticated and user session starts, the login process sets the real UID, file system effective UID and login UID as well as the real GID, effective GID, file system GID and a set of supplemental GIDs for the user trying to log in, and spawns the initial login shell as the first process the user can interact with.

6.6.2 User Identity Changing

Users can change their identity (i.e., switch to another identity) using one of the following commands provided by the TOE:

- su command: The su command is intended for a switch to another identity that establishes a new login session and spawns a new shell with the new identity. When invoking su, the user must provide the credentials associated with the target identity - i.e. when the user wants to switch to another user ID, it has to provide the password protecting the account of the target user. In EulerOS, the capability to login as the root account has been restricted to specified terminals only. In addition, the use of the su command to switch to root has been restricted to users belonging to a special group (*wheel*). Users who don’t have access to a terminal where root login is allowed or are not member of that special group will not be able to switch to root account, even if they would know the authentication information for root.
- sudo command: The sudo command is intended for giving users permissions to execute commands with another user identity (specified by *-u* option). When invoking sudo, the user has to authenticate with his own credentials.

When switching identities, the real, file system and effective user ID and real, file system and effective group ID are changed to the one the user specified in the command (after successful authentication as this user).

6.6.3 Authentication Data Management

Each TOE instance maintains its own set of users with their passwords and attributes. Although the same human user may have accounts on different servers interconnected by a network and running an instance of the TOE, those accounts and their parameter are not synchronized on different TOE instances. As a result the same user may have different user names, different user Ids, different passwords and different attributes on different machines within the networked environment.

Each TOE instance within the network maintains its own administrative database by making all administrative changes on the local TOE instance. System administration has to ensure that all machines within the network are configured in accordance with the requirements defined in this Security Target.

The file */etc/passwd* contains for each user the user's name, the id of the user, an indicator whether the password of the user is valid, the principal group id of the user and other (not security relevant) information. The file */etc/shadow* contains for each user a hash of the user's password, the userid, the time the password was last changed, the expiration time as well as the validity period of the password and some other information that are not subject to the security functions as defined in this Security Target. Users are allowed to change their passwords by using the *passwd* command. This command is able to read and modify the contents of */etc/shadow* for the user's password entry, which would ordinarily be inaccessible to a non-privileged user process. Users are also warned to change their passwords at login time if the password will expire soon, and are prevented from logging in if the password has expired.

Note, only file */etc/shadow* is regarded as the system-wide credential file. File */etc/passwd* contains mappings between user name and uid in the whole system, but the most sensitive info (**password**) is put in file */etc/shadow*. Hence, file */etc/passwd* is world-readable in the TOE, just as in all other UNIX/LINUX releases, leaving very limited info in it to attackers.

Therefore, file */etc/passwd* has little special meaning to the whole system, and then, is NOT regarded as a system-wide credential file.

6.6.4 SSH key-based authentication

In addition to the PAM-based authentication, the OpenSSH server is able to perform a key-based authentication. When a user wants to log in, instead of providing a password, the user applies his SSH key. After a successful verification, the OpenSSH server considers the user as authenticated and performs the PAM-based operations as outlined above.

To establish a key-based authentication, a user first has to generate an RSA or ECDSA key pair. The private part of the key pair remains on the client side, and should be protected safely (the same way as protecting the local user password). The public part is copied to the server into the file *.ssh/authorized_keys* which resides in the home directory of the account to login. When the login operation is performed, the SSHv2 protocol tries to perform the "publickey" authentication using the private key on the client side and the public key found on the server side.

6.6.5 Session locking

The TOE uses the *screen(1)* application which locks the current session of the user either after an administrator-specified time of inactivity or upon the user's request.

To unlock the session, the user must supply his password. *Screen* uses PAM to validate the password and allows the user to access his session after a successful validation.

6.6.6 X.509 Certificate Validation and Generation

Every component in EulerOS that uses X.509 certificates is responsible for performing certificate validation. For example, server certification has to be validated when creating a SSL/TLS connection to a remote server.

EulerOS contains a repository of public CA certificates and will select a certificate based on criteria such as entity name for the communication partner, any extended key usage constraints, and cryptographic algorithms associated with the certificate. EulerOS component will use the same kinds of information along with a certification path and certificate trust lists as part of deciding to accept the certificate.

Following steps are needed to validate a supplied certificate:

- First, build a certificate chain starting from the supplied certificate and ending in the root CA. This is done by looking up the certificate of the issuer of the current certificate. If a certificate is found which is its own issuer, it is assumed to be the root CA. To find an issuer’s certificate, all certificates whose subject name matches the issuer name of the current certificate are tested. The relevant authority key identifier components of the current certificate (if present) must match the subject key identifier (if present) and issuer and serial number of the candidate issuer, in addition the keyUsage extension of the candidate issuer (if present) must permit certificate signing. The lookup first excludes all untrusted or revoked certificates, then search in the trusted certificates. The root CA is always looked up in the trusted certificate list: if the certificate to verify is a root certificate then an exact match must be found in the trusted list. CRL mechanism can be used to check the revocation status of a certificate.
- Second, check every untrusted certificate's extensions for consistency with the supplied purpose. The supplied or "leaf" certificate must have extensions compatible with the supplied purpose and all other certificates must also be valid CA certificates.
- Third, check the trust settings on the root CA. The root CA should be trusted for the supplied purpose. For compatibility, a certificate with no trust settings is considered to be valid for all purposes.
- Last, check the validity of the certificate chain. The validity period is checked against the current system time and the *notBefore* and *notAfter* dates in the certificate. The certificate signatures are also checked at this point.

If all operations above complete successfully, then the supplied certificate is considered valid; otherwise it is regarded as invalid.

If certificate validation fails, or if EulerOS is not able to check the validation status for a certificate, EulerOS will not establish a trusted network channel by default; the user can select to bypass the certificate validation using some options. E.g., option *--no-check-certificate* can lead *wget* to ignore server certificate.

Certification validation for system updates and integrity verification is enabled by default. However, the administrator can bypass the results of a failed certificate validation using special option for *rpm* (--nosignature) or *yum* (--nogpgcheck), although this is not recommended.

When a certificate enrollment request needs to generate, EulerOS will prompt the user to input information bound to the request (including common name), information about the cryptographic algorithms used for the request, any certification extensions, and information about the client requesting the certificate.

6.6.7 SFR Summary

- FIA_AFL.1: After the number of consecutive failed authentication attempts for a user account has been surpassed, EulerOS can lockout the user account according to the configuration file.
- FIA_UAU.5: EulerOS provides authentication using a username and password.
- FIA_X509_EXT.1: EulerOS validates X.509 certificates according to RFC 5280 and provides CRL mechanism for applications to check certificate revocation status.
- FIA_X509_EXT.2: EulerOS uses X.509 certificates for TLS and HTTPS.

6.7 Trusted Path/Channels

EulerOS provides trusted network channels to communicate with supporting IT infrastructure or applications:

- Using TLS (HTTPS) for certificate enrollment; CRL checking; authentication to network resources such as web (HTTPS).
- SSH is for user log in.

6.7.1 Local trusted path

EulerOS provides the mechanism for the administrator to specify the interval of inactivity after which an active session will be terminated by force. Before a user tries to log on to EulerOS from a local console, he can invoke the real login process by pressing the key sequence (Alt+SysRq+K), which is called Secure Attention Key (SAK). The SAK is captured by the TSF and cannot be intercepted or altered by any user process. On receiving the key sequence, the kernel will kill all processes which have opened the console, and the trusted login process will then be restarted by the *init* process, and then the user can login to the system securely. In this way, it can be assured that the login prompt the user can see on the console is actually from the system login process, not from some trojan programs.

6.7.2 Network-based trusted channel

SSH

EulerOS provides the Secure Shell Protocol Version 2 (SSH v2.0) to allow users establish a secure connection from a remote host and perform a logon to the TOE.

The protocol is implemented in OpenSSH application suite. The protocol works in the client-server model, which means that the connection is established by the SSH client trying to connect to the SSH server. The SSH client initiates the connection setup process and uses public key cryptography to verify the identity of the SSH server. After the setup phase, the SSH protocol uses strong symmetric encryption (e.g. AES) and hashing algorithms to ensure the privacy and integrity of the data that is exchanged between the client and server.

TLS

EulerOS also uses Transport Layer Security protocol (successor to SSL) to provide a trusted communication channel between itself and authorized IT entities. The Transport Layer Security protocol aims primarily to provide privacy and data integrity between two communicating computer applications. Secured by TLS, connections between a client and a server have one or more of the following properties:

- The connection is private (or secure) because symmetric cryptography is used to encrypt the data transmitted. The keys for this symmetric encryption are generated uniquely for each connection based on a shared secret negotiated at the start of the session. The server and client negotiate the details about which encryption algorithm and which cryptographic key to use before the first byte of application data is transmitted through the connection. The negotiation of the shared secret is both secure (the negotiated secret is unavailable to eavesdroppers and cannot be obtained, even by an attacker who places themselves in the middle of the connection) and reliable (no attacker can modify the communications during the negotiation without being detected).
- The identity of the communicating parties can be authenticated using public-key cryptography. This authentication can be optional, but is generally required for at least one of the both parties (typically the server).
- The connection ensures integrity of each message transmitted by including an authentication code in the message to prevent undetected loss or alteration of the data during transmission.

6.7.3 SFR Summary

- FTP_ITC_EXT.1: EulerOS provides trusted network channels, which protect data in transit from disclosure, provide data integrity and endpoint identification that is used by TLS, HTTPS and SSH for network-based authentication and certification validation.
- FTP_TRP.1: EulerOS provide a local trusted path service using SAK and a network-based trusted channel built on the network protocols described in this section.

7 Appendix A. SBOP files and rationale.

The rationales provided for each file are classified as the following types:

- Type 1: There are not functions in the file at all.
- Type 2: The ELF file does NOT use character arrays on stack.
- Type 3: The stack protection mechanism is disabled by an explicit attribute 'always_inline’.
- Type 4: The ELF file uses character arrays on stack, but also uses secure functions to operate on the array.
- Type 5: The ELF file is for a special programming language, which has special mechanisms to avoid stack smashing.

FILE	Type
/lib/audit/sotruss-lib.so:FAIL	No array on stack
/lib/gconv/CP1125.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/T.61.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-10.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-FI-SE-A.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM5347.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/VISCII.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM4899.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP1251.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO_6937.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP771.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP1256.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-DK-NO.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/INIS-CYRILLIC.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-ES.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM277.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1133.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM863.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1142.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1143.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1008_420.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP932.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1157.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EUC-KR.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/SHIFT_JISX0213.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/MACINTOSH.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1155.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/INIS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-FI-SE.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM932.so:FAIL	Special attribute: __attribute ((always_inline))

/lib/gconv/SAMI-WS2.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-7.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM278.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO-2022-CN-EXT.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM275.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/JOHAB.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1161.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO_10367-BOX.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM4909.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1047.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IEC_P27-1.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/libGB.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-ES-A.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/GREEK7-OLD.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/libCNS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO_5428.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EUC-JP.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/HP-ROMAN8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1008.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1164.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/BRF.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/MIK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM290.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1148.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP737.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1364.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1026.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/MAC-UK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-UK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM9448.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP1252.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM424.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM437.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM868.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/TIS-620.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM943.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-IT.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/HP-GREEK8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/GBBIG5.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-AT-DE.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM860.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/RK1048.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-6.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM9066.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1129.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/GEORGIAN-ACADEMY.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-4.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM852.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO-IR-197.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/KOI8-T.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM864.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM901.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM4517.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM869.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-11.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/LATIN-GREEK-1.so:FAIL	Special attribute: __attribute ((always_inline))

/lib/gconv/IBM420.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM905.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP770.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-13.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-US.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP1254.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM880.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/libKSC.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO_6937-2.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/HP-THAI8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO_11548-1.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1122.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/INIS-8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM866.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/UTF-16.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM297.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CSN_369103.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1146.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP775.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM939.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/HP-ROMAN9.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM256.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/NATS-DANO.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-9.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1162.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM935.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-16.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP774.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-AT-DE-A.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM875.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM857.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1147.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM865.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM423.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/KOI8-R.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EUC-JP-MS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM922.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1156.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-14.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP1253.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1160.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO_5427-EXT.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM851.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM16804.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/LATIN-GREEK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-5.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP1255.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM871.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/GEORGIAN-PS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM903.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1167.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1158.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1046.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ARMSCII-8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1153.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP773.so:FAIL	Special attribute: __attribute ((always_inline))

/lib/gconv/IBM870.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/MAC-CENTRALEUROPE.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM284.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/GOST_19768-74.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1154.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM855.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM856.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM921.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/KOI-8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CWI.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1163.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM500.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP10007.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ANSI_X3.110.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/TCVN5712-1.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1371.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1132.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM9030.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/libJIS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM930.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP772.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP1258.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/PT154.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/BIG5HKSCS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EUC-JISX0213.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-PT.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM891.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1141.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-2.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM862.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM933.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/UTF-7.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1166.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-IS-FRISS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1123.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-FR.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1112.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/UHC.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1124.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/UTF-32.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/UNICODE.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-CA-FR.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO_5427.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM902.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/GBK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/BIG5.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP1250.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1130.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM861.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1399.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1144.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/NATS-SEFI.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/MAC-SAMI.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/CP1257.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM937.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1140.so:FAIL	Special attribute: __attribute ((always_inline))

/lib/gconv/KOI8-RU.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EUC-CN.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/HP-TURKISH8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM866NAV.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM4971.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM037.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/KOI8-U.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ECMA-CYRILLIC.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM850.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/libISOIR165.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO-2022-KR.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO_2033.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1097.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ASMO_449.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM874.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO-2022-CN.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-3.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-1.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM12712.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM274.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/GB18030.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM280.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-DK-NO-A.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO-2022-JP-3.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1137.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM918.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM273.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1390.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-15.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/TSCII.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1388.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1145.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO-IR-209.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EBCDIC-ES-S.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM285.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO8859-9E.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1004.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/EUC-TW.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/GBGBK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO646.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISIRI-3342.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/SJIS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM904.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/DEC-MCS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/GREEK-CCITT.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM038.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1025.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/libJISX0213.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/GREEK7.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM1149.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/MAC-IS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM281.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/ISO-2022-JP.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/gconv/IBM803.so:FAIL	Special attribute: __attribute ((always_inline))
/lib/grub/i386-pc/iso9660.module:FAIL	No array on stack

/lib/grub/i386-pc/cat.module:FAIL	No array on stack
/lib/grub/i386-pc/hdparm.module:FAIL	No array on stack
/lib/grub/i386-pc/backtrace.module:FAIL	No array on stack
/lib/grub/i386-pc/exfat.module:FAIL	No array on stack
/lib/grub/i386-pc/usbserial_usbdebug.module:FAIL	No array on stack
/lib/grub/i386-pc/linux16.module:FAIL	No array on stack
/lib/grub/i386-pc/play.module:FAIL	No array on stack
/lib/grub/i386-pc/pxeboot.image:FAIL	No array on stack
/lib/grub/i386-pc/gcry_sha256.module:FAIL	No array on stack
/lib/grub/i386-pc/mdraid1x.module:FAIL	No array on stack
/lib/grub/i386-pc/sleep_test.module:FAIL	No array on stack
/lib/grub/i386-pc/part_amiga.module:FAIL	No array on stack
/lib/grub/i386-pc/ufs2.module:FAIL	No array on stack
/lib/grub/i386-pc/video_cirrus.module:FAIL	No array on stack
/lib/grub/i386-pc/usbserial_pl2303.module:FAIL	No array on stack
/lib/grub/i386-pc/squash4.module:FAIL	No array on stack
/lib/grub/i386-pc/xnu.module:FAIL	No array on stack
/lib/grub/i386-pc/memdisk.module:FAIL	No array on stack
/lib/grub/i386-pc/tga.module:FAIL	No array on stack
/lib/grub/i386-pc/ntldr.module:FAIL	No array on stack
/lib/grub/i386-pc/ata.module:FAIL	No array on stack
/lib/grub/i386-pc/testload.module:FAIL	No array on stack
/lib/grub/i386-pc/part_msdos.module:FAIL	No array on stack
/lib/grub/i386-pc/cbtable.module:FAIL	No array on stack
/lib/grub/i386-pc/lsapm.module:FAIL	No array on stack
/lib/grub/i386-pc/cpio.module:FAIL	No array on stack
/lib/grub/i386-pc/affs.module:FAIL	No array on stack
/lib/grub/i386-pc/bsd.module:FAIL	No array on stack
/lib/grub/i386-pc/bitmap_scale.module:FAIL	No array on stack
/lib/grub/i386-pc/usb_keyboard.module:FAIL	No array on stack
/lib/grub/i386-pc/file.module:FAIL	No array on stack
/lib/grub/i386-pc/cmp.module:FAIL	No array on stack
/lib/grub/i386-pc/diskboot.image:FAIL	No array on stack
/lib/grub/i386-pc/progress.module:FAIL	No array on stack
/lib/grub/i386-pc/pxechain.module:FAIL	No array on stack
/lib/grub/i386-pc/part_apple.module:FAIL	No array on stack
/lib/grub/i386-pc/usbserial_ftdi.module:FAIL	No array on stack
/lib/grub/i386-pc/test_blockarg.module:FAIL	No array on stack
/lib/grub/i386-pc/part_plan.module:FAIL	No array on stack
/lib/grub/i386-pc/macbless.module:FAIL	No array on stack
/lib/grub/i386-pc/search_label.module:FAIL	No array on stack
/lib/grub/i386-pc/kernel.exec:FAIL	No array on stack
/lib/grub/i386-pc/trig.module:FAIL	No array on stack
/lib/grub/i386-pc/zfsinfo.module:FAIL	No array on stack
/lib/grub/i386-pc/multiboot2.module:FAIL	No array on stack
/lib/grub/i386-pc/spkmodem.module:FAIL	No array on stack
/lib/grub/i386-pc/extcmd.module:FAIL	No array on stack
/lib/grub/i386-pc/elf.module:FAIL	No array on stack
/lib/grub/i386-pc/search_fs_file.module:FAIL	No array on stack
/lib/grub/i386-pc/pcidump.module:FAIL	No array on stack
/lib/grub/i386-pc/part_dfly.module:FAIL	No array on stack
/lib/grub/i386-pc/raid6rec.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_rsa.module:FAIL	No array on stack
/lib/grub/i386-pc/read.module:FAIL	No array on stack
/lib/grub/i386-pc/minix.module:FAIL	No array on stack
/lib/grub/i386-pc/tftp.module:FAIL	No array on stack

/lib/grub/i386-pc/disk.module:FAIL	No array on stack
/lib/grub/i386-pc/vga.module:FAIL	No array on stack
/lib/grub/i386-pc/setpci.module:FAIL	No array on stack
/lib/grub/i386-pc/video.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_idea.module:FAIL	No array on stack
/lib/grub/i386-pc/cmdline_cat_test.module:FAIL	No array on stack
/lib/grub/i386-pc/legacycfg.module:FAIL	No array on stack
/lib/grub/i386-pc/minix3.module:FAIL	No array on stack
/lib/grub/i386-pc/plan9.module:FAIL	No array on stack
/lib/grub/i386-pc/biosdisk.module:FAIL	No array on stack
/lib/grub/i386-pc/bfs.module:FAIL	No array on stack
/lib/grub/i386-pc/gettext.module:FAIL	No array on stack
/lib/grub/i386-pc/gfxterm_menu.module:FAIL	No array on stack
/lib/grub/i386-pc/cmosdump.module:FAIL	No array on stack
/lib/grub/i386-pc/password.module:FAIL	No array on stack
/lib/grub/i386-pc/lzma_decompress.image:FAIL	No array on stack
/lib/grub/i386-pc/luks.module:FAIL	No array on stack
/lib/grub/i386-pc/cmostest.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_md5.module:FAIL	No array on stack
/lib/grub/i386-pc/tr.module:FAIL	No array on stack
/lib/grub/i386-pc/part_sunpc.module:FAIL	No array on stack
/lib/grub/i386-pc/lsacpi.module:FAIL	No array on stack
/lib/grub/i386-pc/syslinuxcfg.module:FAIL	No array on stack
/lib/grub/i386-pc/part_sun.module:FAIL	No array on stack
/lib/grub/i386-pc/romfs.module:FAIL	No array on stack
/lib/grub/i386-pc/xfs.module:FAIL	No array on stack
/lib/grub/i386-pc/time.module:FAIL	No array on stack
/lib/grub/i386-pc/search.module:FAIL	No array on stack
/lib/grub/i386-pc/macho.module:FAIL	No array on stack
/lib/grub/i386-pc/videoinfo.module:FAIL	No array on stack
/lib/grub/i386-pc/ahci.module:FAIL	No array on stack
/lib/grub/i386-pc/iorw.module:FAIL	No array on stack
/lib/grub/i386-pc/minicmd.module:FAIL	No array on stack
/lib/grub/i386-pc/cbfs.module:FAIL	No array on stack
/lib/grub/i386-pc/help.module:FAIL	No array on stack
/lib/grub/i386-pc/setjmp_test.module:FAIL	No array on stack
/lib/grub/i386-pc/adler32.module:FAIL	No array on stack
/lib/grub/i386-pc/ntfs.module:FAIL	No array on stack
/lib/grub/i386-pc/cryptodisk.module:FAIL	No array on stack
/lib/grub/i386-pc/pxe.module:FAIL	No array on stack
/lib/grub/i386-pc/video_colors.module:FAIL	No array on stack
/lib/grub/i386-pc/ldm.module:FAIL	No array on stack
/lib/grub/i386-pc/gptsync.module:FAIL	No array on stack
/lib/grub/i386-pc/archelp.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_dsa.module:FAIL	No array on stack
/lib/grub/i386-pc/odc.module:FAIL	No array on stack
/lib/grub/i386-pc/cbls.module:FAIL	No array on stack
/lib/grub/i386-pc/bufio.module:FAIL	No array on stack
/lib/grub/i386-pc/gdb.module:FAIL	No array on stack
/lib/grub/i386-pc/sendkey.module:FAIL	No array on stack
/lib/grub/i386-pc/blocklist.module:FAIL	No array on stack
/lib/grub/i386-pc/lnxboot.image:FAIL	No array on stack
/lib/grub/i386-pc/memrw.module:FAIL	No array on stack
/lib/grub/i386-pc/offsetio.module:FAIL	No array on stack
/lib/grub/i386-pc/nilfs2.module:FAIL	No array on stack
/lib/grub/i386-pc/png.module:FAIL	No array on stack

/lib/grub/i386-pc/video_bochs.module:FAIL	No array on stack
/lib/grub/i386-pc/msdospart.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_rmd160.module:FAIL	No array on stack
/lib/grub/i386-pc/minix2_be.module:FAIL	No array on stack
/lib/grub/i386-pc/probe.module:FAIL	No array on stack
/lib/grub/i386-pc/verify.module:FAIL	No array on stack
/lib/grub/i386-pc/cpio_be.module:FAIL	No array on stack
/lib/grub/i386-pc/boot_hybrid.image:FAIL	No array on stack
/lib/grub/i386-pc/true.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_arcfour.module:FAIL	No array on stack
/lib/grub/i386-pc/nativedisk.module:FAIL	No array on stack
/lib/grub/i386-pc/datehook.module:FAIL	No array on stack
/lib/grub/i386-pc/legacy_password_test.module:FAIL	No array on stack
/lib/grub/i386-pc/minix3_be.module:FAIL	No array on stack
/lib/grub/i386-pc/crypto.module:FAIL	No array on stack
/lib/grub/i386-pc/fshelp.module:FAIL	No array on stack
/lib/grub/i386-pc/ufs1_be.module:FAIL	No array on stack
/lib/grub/i386-pc/freedos.module:FAIL	No array on stack
/lib/grub/i386-pc/zfs.module:FAIL	No array on stack
/lib/grub/i386-pc/all_video.module:FAIL	No array on stack
/lib/grub/i386-pc/crc64.module:FAIL	No array on stack
/lib/grub/i386-pc/terminfo.module:FAIL	No array on stack
/lib/grub/i386-pc/udf.module:FAIL	No array on stack
/lib/grub/i386-pc/date.module:FAIL	No array on stack
/lib/grub/i386-pc/diskfilter.module:FAIL	No array on stack
/lib/grub/i386-pc/videotest_checksum.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_camellia.module:FAIL	No array on stack
/lib/grub/i386-pc/configfile.module:FAIL	No array on stack
/lib/grub/i386-pc/mdraid09.module:FAIL	No array on stack
/lib/grub/i386-pc/blscfg.module:FAIL	No array on stack
/lib/grub/i386-pc/gfxterm.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_des.module:FAIL	No array on stack
/lib/grub/i386-pc/boot.image:FAIL	No array on stack
/lib/grub/i386-pc/aout.module:FAIL	No array on stack
/lib/grub/i386-pc/tar.module:FAIL	No array on stack
/lib/grub/i386-pc/efiemu.module:FAIL	No array on stack
/lib/grub/i386-pc/halt.module:FAIL	No array on stack
/lib/grub/i386-pc/afs.module:FAIL	No array on stack
/lib/grub/i386-pc/test.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_whirlpool.module:FAIL	No array on stack
/lib/grub/i386-pc/font.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_seed.module:FAIL	No array on stack
/lib/grub/i386-pc/cdboot.image:FAIL	No array on stack
/lib/grub/i386-pc/video_fb.module:FAIL	No array on stack
/lib/grub/i386-pc/echo.module:FAIL	No array on stack
/lib/grub/i386-pc/ext2.module:FAIL	No array on stack
/lib/grub/i386-pc/hfsplus.module:FAIL	No array on stack
/lib/grub/i386-pc/ehci.module:FAIL	No array on stack
/lib/grub/i386-pc/hfs.module:FAIL	No array on stack
/lib/grub/i386-pc/serial.module:FAIL	No array on stack
/lib/grub/i386-pc/part_gpt.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_rfc2268.module:FAIL	No array on stack
/lib/grub/i386-pc/minix_be.module:FAIL	No array on stack
/lib/grub/i386-pc/signature_test.module:FAIL	No array on stack

/lib/grub/i386-pc/newc.module:FAIL	No array on stack
/lib/grub/i386-pc/cbtime.module:FAIL	No array on stack
/lib/grub/i386-pc/drivemap.module:FAIL	No array on stack
/lib/grub/i386-pc/usb.module:FAIL	No array on stack
/lib/grub/i386-pc/ls.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_rijndael.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_md4.module:FAIL	No array on stack
/lib/grub/i386-pc/part_dvh.module:FAIL	No array on stack
/lib/grub/i386-pc/terminal.module:FAIL	No array on stack
/lib/grub/i386-pc/cpuid.module:FAIL	No array on stack
/lib/grub/i386-pc/lvm.module:FAIL	No array on stack
/lib/grub/i386-pc/keystatus.module:FAIL	No array on stack
/lib/grub/i386-pc/hfspluscomp.module:FAIL	No array on stack
/lib/grub/i386-pc/normal.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_blowfish.module:FAIL	No array on stack
/lib/grub/i386-pc/uhci.module:FAIL	No array on stack
/lib/grub/i386-pc/priority_queue.module:FAIL	No array on stack
/lib/grub/i386-pc/net.module:FAIL	No array on stack
/lib/grub/i386-pc/jfs.module:FAIL	No array on stack
/lib/grub/i386-pc/lzopio.module:FAIL	No array on stack
/lib/grub/i386-pc/gzio.module:FAIL	No array on stack
/lib/grub/i386-pc/raid5rec.module:FAIL	No array on stack
/lib/grub/i386-pc/exfctest.module:FAIL	No array on stack
/lib/grub/i386-pc/mmap.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_twofish.module:FAIL	No array on stack
/lib/grub/i386-pc/usbtest.module:FAIL	No array on stack
/lib/grub/i386-pc/vga_text.module:FAIL	No array on stack
/lib/grub/i386-pc/gfxterm_background.module:FAIL	No array on stack
/lib/grub/i386-pc/hello.module:FAIL	No array on stack
/lib/grub/i386-pc/xnu_uuid.module:FAIL	No array on stack
/lib/grub/i386-pc/setjmp.module:FAIL	No array on stack
/lib/grub/i386-pc/pbkdf2.module:FAIL	No array on stack
/lib/grub/i386-pc/pata.module:FAIL	No array on stack
/lib/grub/i386-pc/fat.module:FAIL	No array on stack
/lib/grub/i386-pc/gfxmenu.module:FAIL	No array on stack
/lib/grub/i386-pc/acpi.module:FAIL	No array on stack
/lib/grub/i386-pc/pbkdf2_test.module:FAIL	No array on stack
/lib/grub/i386-pc/lsmmap.module:FAIL	No array on stack
/lib/grub/i386-pc/linux.module:FAIL	No array on stack
/lib/grub/i386-pc/part_acorn.module:FAIL	No array on stack
/lib/grub/i386-pc/btrfs.module:FAIL	No array on stack
/lib/grub/i386-pc/boot.module:FAIL	No array on stack
/lib/grub/i386-pc/pci.module:FAIL	No array on stack
/lib/grub/i386-pc/scsi.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_cast5.module:FAIL	No array on stack
/lib/grub/i386-pc/at_keyboard.module:FAIL	No array on stack
/lib/grub/i386-pc/functional_test.module:FAIL	No array on stack
/lib/grub/i386-pc/password_pbkdf2.module:FAIL	No array on stack
/lib/grub/i386-pc/datetime.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_crc.module:FAIL	No array on stack
/lib/grub/i386-pc/dm_nv.module:FAIL	No array on stack
/lib/grub/i386-pc/sfs.module:FAIL	No array on stack
/lib/grub/i386-pc/chain.module:FAIL	No array on stack
/lib/grub/i386-pc/relocator.module:FAIL	No array on stack
/lib/grub/i386-pc/reboot.module:FAIL	No array on stack

/lib/grub/i386-pc/http.module:FAIL	No array on stack
/lib/grub/i386-pc/ohci.module:FAIL	No array on stack
/lib/grub/i386-pc/geli.module:FAIL	No array on stack
/lib/grub/i386-pc/hexdump.module:FAIL	No array on stack
/lib/grub/i386-pc/videotest.module:FAIL	No array on stack
/lib/grub/i386-pc/parttool.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_serpent.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_tiger.module:FAIL	No array on stack
/lib/grub/i386-pc/keylayouts.module:FAIL	No array on stack
/lib/grub/i386-pc/usbms.module:FAIL	No array on stack
/lib/grub/i386-pc/morse.module:FAIL	No array on stack
/lib/grub/i386-pc/minix2.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_sha512.module:FAIL	No array on stack
/lib/grub/i386-pc/vbe.module:FAIL	No array on stack
/lib/grub/i386-pc/jpeg.module:FAIL	No array on stack
/lib/grub/i386-pc/lspci.module:FAIL	No array on stack
/lib/grub/i386-pc/ntfscomp.module:FAIL	No array on stack
/lib/grub/i386-pc/hashsum.module:FAIL	No array on stack
/lib/grub/i386-pc/mpi.module:FAIL	No array on stack
/lib/grub/i386-pc/testspeed.module:FAIL	No array on stack
/lib/grub/i386-pc/xnu_uuid_test.module:FAIL	No array on stack
/lib/grub/i386-pc/reiserfs.module:FAIL	No array on stack
/lib/grub/i386-pc/div_test.module:FAIL	No array on stack
/lib/grub/i386-pc/mdraid09_be.module:FAIL	No array on stack
/lib/grub/i386-pc/cs5536.module:FAIL	No array on stack
/lib/grub/i386-pc/gcry_sha1.module:FAIL	No array on stack
/lib/grub/i386-pc/cbmemc.module:FAIL	No array on stack
/lib/grub/i386-pc/eval.module:FAIL	No array on stack
/lib/grub/i386-pc/xzio.module:FAIL	No array on stack
/lib/grub/i386-pc/sleep.module:FAIL	No array on stack
/lib/grub/i386-pc/mda_text.module:FAIL	No array on stack
/lib/grub/i386-pc/loadenv.module:FAIL	No array on stack
/lib/grub/i386-pc/bitmap.module:FAIL	No array on stack
/lib/grub/i386-pc/ufs1.module:FAIL	No array on stack
/lib/grub/i386-pc/regexp.module:FAIL	No array on stack
/lib/grub/i386-pc/zfscrypt.module:FAIL	No array on stack
/lib/grub/i386-pc/search_fs_uuid.module:FAIL	No array on stack
/lib/grub/i386-pc/truecrypt.module:FAIL	No array on stack
/lib/grub/i386-pc/loopback.module:FAIL	No array on stack
/lib/grub/i386-pc/part_bsd.module:FAIL	No array on stack
/lib/grub/i386-pc/procfs.module:FAIL	No array on stack
/lib/grub/i386-pc/usbserial_common.module:FAIL	No array on stack
/lib/grub/i386-pc/multiboot.module:FAIL	No array on stack
/lib/i686/nosegneg/libpthread-2.17.so:FAIL	No array on stack
/lib/i686/nosegneg/librt-2.17.so:FAIL	No array on stack
/lib/i686/nosegneg/libthread_db-1.0.so:FAIL	No array on stack
/lib/i686/nosegneg/libc-2.17.so:FAIL	No array on stack
/lib/i686/nosegneg/libm-2.17.so:FAIL	No array on stack
/lib/ld-2.17.so:FAIL	No array on stack
/lib/libanl-2.17.so:FAIL	No array on stack
/lib/libBrokenLocale-2.17.so:FAIL	No array on stack
/lib/libc-2.17.so:FAIL	No array on stack
/lib/libcidn-2.17.so:FAIL	No array on stack
/lib/libcrypt-2.17.so:FAIL	No array on stack
/lib/libdl-2.17.so:FAIL	No array on stack
/lib/libgcc_s-4.8.5-20150702.so.1:FAIL	No array on stack

/lib/libm-2.17.so:FAIL	No array on stack
/lib/libmemusage.so:FAIL	No array on stack
/lib/libnsl-2.17.so:FAIL	No array on stack
/lib/libnss_compat-2.17.so:FAIL	No array on stack
/lib/libnss_db-2.17.so:FAIL	No array on stack
/lib/libnss_dns-2.17.so:FAIL	No array on stack
/lib/libnss_files-2.17.so:FAIL	No array on stack
/lib/libnss_hesiod-2.17.so:FAIL	No array on stack
/lib/libnss_nis-2.17.so:FAIL	No array on stack
/lib/libnss_nisplus-2.17.so:FAIL	No array on stack
/lib/libpcprofile.so:FAIL	No character array on stack
/lib/libpthread-2.17.so:FAIL	No array on stack
/lib/librt-2.17.so:FAIL	No character array on stack
/lib/libSegFault.so:FAIL	No array on stack
/lib/libthread_db-1.0.so:FAIL	No array on stack
/lib/libutil-2.17.so:FAIL	No array on stack
/lib/modules/3.10.0-327.53.58.73.h3.x86_64/vdso/vdso32-syscall.so:FAIL	No array on stack
/lib/modules/3.10.0-327.53.58.73.h3.x86_64/vdso/vdso.so:FAIL	No array on stack
/lib/modules/3.10.0-327.53.58.73.h3.x86_64/vdso/vdso32-int80.so:FAIL	No array on stack
/lib/modules/3.10.0-327.53.58.73.h3.x86_64/vdso/vdso32-sysenter.so:FAIL	No array on stack
/lib/rtkaio/i686/nosegneg/librtkaio-2.17.so:FAIL	No array on stack
/lib/rtkaio/librtkaio-2.17.so:FAIL	No array on stack
/lib64/audit/sotruss-lib.so:FAIL	No array on stack
/lib64/gconv/CP1125.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/T.61.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-10.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-FI-SE-A.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM5347.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/VISCII.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM4899.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP1251.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO_6937.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP771.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP1256.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-DK-NO.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/INIS-CYRILLIC.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-ES.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM277.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1133.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM863.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1142.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1143.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1008_420.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP932.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1157.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EUC-KR.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/SHIFT_JISX0213.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/MACINTOSH.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1155.so:FAIL	Special attribute: __attribute ((always_inline))

/lib64/gconv/INIS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-FI-SE.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM932.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/SAMI-WS2.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-7.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM278.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO-2022-CN-EXT.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM275.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/JOHAB.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1161.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO_10367-BOX.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM4909.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1047.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IEC_P27-1.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/libGB.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-ES-A.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/GREEK7-OLD.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/libCNS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO_5428.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EUC-JP.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/HP-ROMAN8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1008.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1164.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/BRF.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/MIK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM290.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1148.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP737.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1364.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1026.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/MAC-UK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-UK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM9448.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP1252.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM424.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM437.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM868.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/TIS-620.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM943.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-IT.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/HP-GREEK8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/GBBIG5.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-AT-DE.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM860.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/RK1048.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-6.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM9066.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1129.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/GEORGIAN-ACADEMY.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-4.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM852.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO-IR-197.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/KOI8-T.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM864.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM901.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM4517.so:FAIL	Special attribute: __attribute ((always_inline))

/lib64/gconv/IBM869.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-11.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/LATIN-GREEK-1.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM420.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM905.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP770.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-13.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-US.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP1254.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM880.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/libKSC.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO_6937-2.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/HP-THAI8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO_11548-1.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1122.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/INIS-8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM866.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/UTF-16.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM297.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CSN_369103.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1146.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP775.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM939.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/HP-ROMAN9.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM256.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/NATS-DANO.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-9.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1162.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM935.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-16.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP774.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-AT-DE-A.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM875.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM857.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1147.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM865.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM423.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/KOI8-R.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EUC-JP-MS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM922.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1156.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-14.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP1253.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1160.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO_5427-EXT.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM851.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM16804.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/LATIN-GREEK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-5.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP1255.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM871.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/GEORGIAN-PS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM903.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1167.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1158.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1046.so:FAIL	Special attribute: __attribute ((always_inline))

/lib64/gconv/ARMSCII-8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1153.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP773.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM870.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/MAC-CENTRALEUROPE.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM284.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/GOST_19768-74.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1154.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM855.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM856.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM921.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/KOI-8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CWI.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1163.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM500.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP10007.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ANSI_X3.110.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/TCVN5712-1.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1371.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1132.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM9030.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/libJIS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM930.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP772.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP1258.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/PT154.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/BIG5HKSCS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EUC-JISX0213.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-PT.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM891.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1141.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-2.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM862.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM933.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/UTF-7.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1166.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-IS-FRISS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1123.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-FR.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1112.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/UHC.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1124.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/UTF-32.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/UNICODE.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-CA-FR.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO_5427.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM902.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/GBK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/BIG5.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/CP1250.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1130.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM861.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1399.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1144.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/NATS-SEFI.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/MAC-SAMI.so:FAIL	Special attribute: __attribute ((always_inline))

/lib64/gconv/CP1257.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM937.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1140.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/KOI8-RU.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EUC-CN.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/HP-TURKISH8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM866NAV.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM4971.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM037.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/KOI8-U.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ECMA-CYRILLIC.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM850.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/libISOIR165.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO-2022-KR.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO_2033.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1097.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ASMO_449.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM874.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO-2022-CN.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-3.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-1.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-8.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM12712.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM274.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/GB18030.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM280.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-DK-NO-A.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO-2022-JP-3.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1137.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM918.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM273.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1390.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-15.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/TSCII.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1388.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1145.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO-IR-209.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EBCDIC-ES-S.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM285.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO8859-9E.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1004.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/EUC-TW.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/GBGBK.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISO646.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/ISIRI-3342.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/SJIS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM904.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/DEC-MCS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/GREEK-CCITT.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM038.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1025.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/libJISX0213.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/GREEK7.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM1149.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/MAC-IS.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM281.so:FAIL	Special attribute: __attribute ((always_inline))

/lib64/gconv/ISO-2022-JP.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gconv/IBM803.so:FAIL	Special attribute: __attribute ((always_inline))
/lib64/gettext/hostname:FAIL	No array on stack
/lib64/ld-2.17.so:FAIL	No array on stack
/lib64/libanl-2.17.so:FAIL	No character array on stack
/lib64/libBrokenLocale-2.17.so:FAIL	No array on stack
/lib64/libc-2.17.so:FAIL	No array on stack
/lib64/libcidn-2.17.so:FAIL	No array on stack
/lib64/libcrypt-2.17.so:FAIL	No character array on stack
/lib64/libdl-2.17.so:FAIL	No character array on stack
/lib64/libgcc_s-4.8.5-20150702.so.1:FAIL	No array on stack
/lib64/libGLESv2.so.2.0.0:FAIL	No character array on stack
/lib64/libgraphite2.so.3.0.1:FAIL	No array on stack
/lib64/libicudata.so.50.1.2:FAIL	No character array on stack
/lib64/libiptc.so.0.0.0:FAIL	Containing char array, but protected using fgets() or incoming data len is always checked
/lib64/libjson.so.0.1.0:FAIL	Long char array for printing integers, safe.
/lib64/libm-2.17.so:FAIL	No array on stack
/lib64/libmemusage.so:FAIL	No array on stack
/lib64/libnsl-2.17.so:FAIL	No array on stack
/lib64/libnss_compat-2.17.so:FAIL	No array on stack
/lib64/libnss_db-2.17.so:FAIL	No array on stack
/lib64/libnss_dns-2.17.so:FAIL	No array on stack
/lib64/libnss_files-2.17.so:FAIL	No array on stack
/lib64/libnss_hesiod-2.17.so:FAIL	No array on stack
/lib64/libnss_nis-2.17.so:FAIL	No array on stack
/lib64/libnss_nisplus-2.17.so:FAIL	No array on stack
/lib64/libpcprofile.so:FAIL	No array on stack
/lib64/libpthread-2.17.so:FAIL	No array on stack
/lib64/librt-2.17.so:FAIL	No array on stack
/lib64/libSegFault.so:FAIL	No array on stack
/lib64/libthread_db-1.0.so:FAIL	No array on stack
/lib64/libungif.so.4.1.6:FAIL	No character array on stack
/lib64/libutil-2.17.so:FAIL	No array on stack
/lib64/rtkaio/librtkaio-2.17.so:FAIL	No character array on stack
/usr/bin/checksctp:FAIL	No character array on stack
/usr/bin/clibrary:FAIL	No array on stack
/usr/bin/clibrary2:FAIL	No array on stack
/usr/bin/gencat:FAIL	No character array on stack
/usr/bin/getconf:FAIL	No array on stack
/usr/bin/getent:FAIL	No array on stack
/usr/bin/gjs-console:FAIL	character array is protected by fgets()
/usr/bin/iconv:FAIL	No character array on stack
/usr/bin/ldns-chaos:FAIL	No character array on stack
/usr/bin/ldns-key2ds:FAIL	No character array on stack
/usr/bin/ldns-read-zone:FAIL	No array on stack
/usr/bin/ldns-update:FAIL	No array on stack
/usr/bin/locale:FAIL	No array on stack
/usr/bin/localedef:FAIL	No array on stack
/usr/bin/makedb:FAIL	No character array on stack
/usr/bin/mkrfc2734:FAIL	No character array on stack
/usr/bin/msgcat:FAIL	No array on stack
/usr/bin/msgcomm:FAIL	No array on stack
/usr/bin/msgconv:FAIL	No array on stack
/usr/bin/msgen:FAIL	No array on stack
/usr/bin/msgexec:FAIL	No array on stack

/usr/bin/msguniq:FAIL	No array on stack
/usr/bin/pldd:FAIL	No array on stack
/usr/bin/ppdhtml:FAIL	No array on stack
/usr/bin/ppdi:FAIL	No array on stack
/usr/bin/ppdpo:FAIL	No array on stack
/usr/bin/rpcgen:FAIL	No array on stack
/usr/bin/rpmdumpheader:FAIL	No character array on stack
/usr/bin/sdp_long_message:FAIL	No array on stack
/usr/bin/sprof:FAIL	No array on stack
/usr/bin/xsetmode:FAIL	No array on stack
/usr/sbin/build-locale-archive:FAIL	No array on stack
/usr/sbin/capsh:FAIL	Character assay on stack, protected by fgets()
/usr/sbin/dump_mem_compress:FAIL	No character array on stack
/usr/sbin/dump_mem_imag:FAIL	No character array on stack
/usr/sbin/dump_mem_uncompress:FAIL	No character array on stack
/usr/sbin/glibc_post_upgrade.i686:FAIL	Character array on stack is used as data array, protected by length check
/usr/sbin/glibc_post_upgrade.x86_64:FAIL	No array on stack
/usr/sbin/iconvconfig:FAIL	No array on stack
/usr/sbin/iconvconfig.i686:FAIL	No array on stack
/usr/sbin/iconvconfig.x86_64:FAIL	No array on stack
/usr/sbin/ldconfig:FAIL	No array on stack
/usr/sbin/load_policy:FAIL	No array on stack
/usr/sbin/lockdev:FAIL	Character array on stack are protected by length limit (MAXPATHLEN+1)
/usr/sbin/mem_show:FAIL	No character array on stack
/usr/sbin/mklost+found:FAIL	Character array on stack is protected by length check.
/usr/sbin/nologin:FAIL	Character array on stack are protected by length check (red())
/usr/sbin/parse_err:FAIL	No array on stack
/usr/sbin/setcap:FAIL	Character array on stack are protected by length limit (far more tahn needed)
/usr/sbin/sln:FAIL	No character array on stack
/usr/sbin/tickadj:FAIL	No array on stack
/usr/sbin/zdump:FAIL	No array on stack
/usr/sbin/zic:FAIL	No array on stack

A. Acronyms

Acronym	Meaning
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
ASLR	Address Space Layout Randomization
CEM	Common Evaluation Methodology for Information Technology Security - Evaluation Methodology, CCMB-2012-09-004, Version 3.1, Revision 4, September 2012.
CESG	Communications-Electronics Security Group
CMC	Certificate Management over CMS
CMS	Cryptographic Message Syntax
CN	Common Names
CRL	Certificate Revocation List
CSA	Computer Security Act

DAC	Discretionary Access Control
DEP	Data Execution Prevention
DES	Data Encryption Standard
DHE	Diffie-Hellman Ephemeral
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
DT	Date/Time Vector
DTLS	Datagram Transport Layer Security
EAP	Extensible Authentication Protocol
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECDSA	Elliptic Curve Digital Signature Algorithm
EST	Enrollment over Secure Transport
FIPS	Federal Information Processing Standards
HMAC	Hash-based Message Authentication Code

HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPA	Identity, Policy, and Audit
ISO	International Organization for Standardization
IT	Information Technology
ITSEF	Information Technology Security Evaluation Facility
LUKS	Linux Unified Key Setup
NFC	Near Field Communication
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NSS	Network Security Services
OCSP	Online Certificate Status Protocol
OID	Object Identifier

OMB	Office of Management and Budget
OS	Operating System
PII	Personally Identifiable Information
PKI	Public Key Infrastructure
PP	Protection Profile
RBG	Random Bit Generator
RFC	Request for Comment
RNG	Random Number Generator
RNGVS	Random Number Generator Validation System
SAN	Subject Alternative Name
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SIP	Session Initiation Protocol
S/MIME	Secure/Multi-purpose Internet Mail Extensions

SSSD	System Security Services Daemon
SWID	Software Identification
TLS	Transport Layer Security
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USB	Universal Serial Bus
XCCDF	eXtensible Configuration Checklist Description Format
XOR	Exclusive OR