WCPP V0.1 L5 EXTERNAL INTERACT DEFINITION

[TOP SECRET]

Acronyms

ACRONYM	DEFINITION
WCPP	WhyCard Proprietary Protocol
ID	Identificator
UTC	Coordinated Universal Time
SSL	Secure Sockets Laye
HTTP	The Hypertext Transfer Protocol
HTTPS	The Hypertext Transfer Protocol Secure
WCPPQH	WCPP Query Headers
WCPPRH	WCPP Response Headers
IMEI	International Mobile Equipment Identity
MAC	Medium Access Control
AD	Authentication Data

Table of Contents

L1 Interaction

Security Notice

- L5 uses SSL encryption, signed by well-known distributor
- L5 should disable previous AD, if the new one is given
- L5 should allow the device to be changed only once a week, w/o Admistrator’s permission

Endpoints

WCPP V0.1 Type 0

L1.0.Authorisation

This endpoint should be called from L1 without authorisation.

L1 sends packet directly to L5, which does checks of WCPP Version According to rules, described in PD1001 [1], after that authorisates user or not, according to input data. If authorises, L5 responds with AD.

L4 Interaction

Security Notice

- L5 uses SSL encryption, signed by well-known distributor
- L5 should remember the last token used by each L4.
- L5 should raise a critical security warning if there was found usage of an old Authorisation/Backup Key or wrong Initialization Token
- L5 should raise a non-critical security warning if there was any other inapropriate usage of protocol.

Endpoints

L5.Preauthorisation.1

This section is about endpoint which is not under document's scope. Included only just for information.

This endpoint should be called from Administrator Side.

L5 creates a short time (about 1 hour) 64 bytes key, that we call Initialization Token, based on given device ID. This action should be called only by users with technical administrators' righs.

L4.Preauthorisation.2

This endpoint should be called from L4 without authorisation.

The key generated with *L5.Preauthorisation.1* should be inserted into L4 device and device should call an *L4.Preauthorisation.2* at L5 via HTTPS, giving its device ID and Initialization Token. L5 should return two unique 128 bytes keys:

- Authorisation Key
- Backup Key

Those keys are designed to be stored at both L4 and L5 devices.

L4.Packet Transfer

This endpoint should be authorised with Authorisation Key.

Simply, transferring L1 => L5 packets.

L4.Ping

This endpoint could be authorised with both of Authorisation Key and Backup Key.

L4 asks L5 about it current authorisation status.

Authorisation

Packets should be send via HTTPS with Authorisation header containing either L4 Authorisation Key or L4 Backup Key. The type of key is not provided as it is expected that L5 has that keys stored in database.

With Authorisation Key

This method is preffered and used in most of cases. After each request authorised with Authorisation Key, server should include in headers the new Authorisation Key (*W-New-Authorisation-Key*) to be used in future. L4's request could include header *W-Ask-New-Backup-Key* with value of 1. In that case, the new Backup Key should also be included in L5's response (*W-New-Backup-Key*).

With Backup Key

This method is used only as fallback method for queries of type *L4.Ping*. Possible scenarios is to use it after previous response transfer fail with unknown error, or the response was **401** *UNAUTHORISED*. L5's response should include new Authorisation Key and Backup Key. L4 should try to use Backup Key until it got error **401** *UNAUTHORISED* or a new keypair. If it gets **401** error, the usage of L4 is being blocked until it gets a new Initialization Token inserted.

WCPP Headers

WCPP V0.1 Request Headers (shortly WCPPQH V0.1)

Name	*Type*	*Optional*	*Version Dependent*	*Description*
W-Major-Version	8-bit integer	False	False	WCPP Major Version
W-Minor-Version	8-bit integer	False	False	WCPP Minor Version
W-Authorisation	String (HEX, 256 characters)	True	True	WCPP Authorisation
W-Ask-New-Backup-Key	Boolean	True	True	L4 asks L5 about new keypair. Could be used only on Ping request
W-Init-Token	String (HEX, 128 characters)	True	True	Initialization request from L4

WCPP V0.1 Response Headers (shortly WCPPRH V0.1)

Name	*Type*	*Optional*	*Description*
W-New-Authorisation-Key	String (HEX, 256 characters)	False	New Authorisation Key
W-New-Backup-Key	String (HEX, 256 characters)	True	New Backup Key

OK Payload

L5.Preauthorisation.1

This section is about endpoint which is not under document's scope. Included only just for information.

Request

Headers

Default HTTPS Headers.

Body

Name	*Type*	*Optional*	*Description*
deviceID	String (HEX, 32 characters)	False	L4 Device ID

Response

Headers

Default HTTPS Headers.

Body

Name	*Type*	*Optional*	*Description*
initializationToken	String (HEX, 128 characters)	False	L4 Initialization Token
expires	64-bit integer	True	Initialization Token expire time

L4.Preauthorisation.2

Request

Headers

WCPPQH V0.1 with *W-Init-Token*.

Body

Empty body.

Response

Headers

WCPPRH V0.1 with *W-New-Backup-Key*.

Body

Empty Body.

L4.Ping

Request

Headers

1. WCPPQH V0.1 with *W-Authorisation-Key*.
2. WCPPQH V0.1 with *W-Authorisation-Key* and *W-Ask-New-Backup-Key*.
3. WCPPQH V0.1 with *W-Backup-Key*.

Body

Empty body.

Response

Headers

1. WCPPRH V0.1.
2. WCPPRH V0.1 with *W-New-Backup-Key*.
3. WCPPRH V0.1 with *W-New-Backup-Key*.

Body

Empty Body.

L4.Packet Transfer

Request

Headers

WCPPQH V0.1 with *W-Authorisation*

Body

WCPP V0.1 Packet.

Response

Headers

WCPPRH V0.1

Body

Type-dependent response.

L1.0.Authorisation

Request

Headers

WCPPQH V0.1

Body

Name	*Type*	*Optional*	*Description*
email	String (var. length)	False	User email
password	String (HEX, 64 characters)	False	User password, encrypted with SHA-256
nfcMac	String (HEX, 12 characters)	True	MAC-Address of NFC module in L1
Imei	64-bit integer	True	IMEI number of L1

Response

Headers

WCPPRH V0.1

Body

Name	*Type*	*Optional*	*Description*
accessToken	String (HEX, 256 characters)	False	AD in order to authentification in future

Error Payload

Name	*Type*	*Optional*	*Description*
errorCode	32-bit integer	False	Internal error code. Should be presented. Could be -1 to comply with security restrictions, but L5 should store the real error code in logs.
errorMessage	String	True	Error description, based on error code. Could include additional data.
errorDescription	Object	True	For some requests, could be used to transfer additional data.

HTTP Statuses to be Used

Code	*Ref. Code*	*Description*
200	OK	Processed in default way, without any errors.
400	BAD_REQUEST	The request was malformed, parameters were incorrect. (a)
401	UNAUTHORISED	The request was provided with wrong authorisation data. (a)
403	ACCESS_DENIED	The request was rejected for any other security reason. For example: IP got blocked. (a)
404	NOT_FOUND	The request asked for inapropriate endpoint.
429	REQUEST_TROTTLED	Client made too much requests. Appliable only for requests w/o authorisation.
500	INTERNAL_ERROR	Unclassified error during processing request. (a)
501	NOT_IMPLEMENTED	Should be used as response to requests with wrong version. (a)
503	NOT_AVAILABLE	L5 is currently down due to technical reasons (maintenance for example).

(a) – Response must include Error Payload as described above.

WCPP V0.1 Processing

L5 should receive packet and do the following checks:

- WCPPVB Check (0 and 1 bytes), as described in PD1001 [1], if fail – return **501**
- Request Type (2 byte), should be acceptable for this version, if fail – return **400**
- Checksum (from 3 byte to 35th excluding) – should be proper SHA-256 checksum for Payload block (starting with 35th byte to the end of Payload block), if fail – return **400**
- Signature Check: L1 and L4 device IDs should be recognizable by the L5, if fail – return **403**

WCPP V0.1 Payload Processing

WCPP V0.1 Type 0

L5 should unwrap the Payload and do the following:

- Timestamp Check: `1 <= Current Timestamp - Packet Timestamp <= 10`, if fail – return **400**
- MAC-Address Check: MAC-Address of L1 should be the same as in previous requests, if fail – return **403**
    - MAC address should always be presented
- IMEI Check: IMEI of L1 should be the same as in previous requests, if fail – return **403**
    - If all previous requests were w/o IMEI, L5 should store the new IMEI and do checks in future with it
- RFU Check: RFU bytes should be equal to 0, if fail – return **400**
- AD Check: AD should be recognizable by L5, should be active and should be requested by the proper device, if fail – return **401**

References

1. PD1001