Update Server

External Specification

**Version 1.2**

**Jacky Gu**

**Jacky.gu@titan-arc.com**

**Table of Contents**

[0 Document Information 4](#_Toc340604639)

[0.1 Revision History 4](#_Toc340604640)

[0.2 Reference 4](#_Toc340604641)

[1 Architecture 5](#_Toc340604642)

[1.1 Master Server 5](#_Toc340604643)

[1.2 Slave Server 5](#_Toc340604644)

[2 Action Flow 6](#_Toc340604645)

[2.1 Account Management 6](#_Toc340604646)

[2.2 Device Type / Service Type Management 6](#_Toc340604647)

[2.3 File management 6](#_Toc340604648)

[3 GUI Layer 8](#_Toc340604649)

[3.1 Signature List 8](#_Toc340604650)

[3.2 Add / Modify Panel 8](#_Toc340604651)

[3.3 Log Management 9](#_Toc340604652)

[3.4 Slave Server List 9](#_Toc340604653)

[3.5 System Configuration Panel 🡪 Auto Sync 10](#_Toc340604654)

[3.6 GUI Modification 10](#_Toc340604655)

[4 Signature Update / Download 12](#_Toc340604656)

[4.1 Background 12](#_Toc340604657)

[4.2 Flow 12](#_Toc340604658)

[4.3 API 13](#_Toc340604659)

[4.4 Full-Incremental 14](#_Toc340604660)

[Example 14](#_Toc340604661)

[4.5 Auto Upgrade Schedule Arrangement 15](#_Toc340604662)

[5 Firmware Update / Download 16](#_Toc340604663)

[5.1 Flow 16](#_Toc340604664)

[5.2 API 16](#_Toc340604665)

[5.3 Firmware Version Format 18](#_Toc340604666)

[5.4 Auto Upgrade Schedule Arrangement 19](#_Toc340604667)

[6 Token Mechanism 20](#_Toc340604668)

[6.1 Generate Token 20](#_Toc340604669)

[6.2 Token- file Mapping 20](#_Toc340604670)

[6.3 Download 20](#_Toc340604671)

# Document Information

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** |
| 2012/11/8 | 1.0 | Jacky Gu | Initial document. |
| 2012/11/10 | 1.1 | Jacky Gu | 1. Add firmware upgrade definition; 2. Update download API definition(shorten URL, add response for invalid token); 3. Update token mechanism; |
| 2012/11/13 | 1.2 | Jacky Gu | 1. Update Device Type/Service Type Management; 2. High light “default upgrade time”; 3. Add description of firmware version format; |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Reference

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Document** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Architecture

## Master Server

The Master Server manages signatures. We add/delete/update signatures on Master Server. Then Master Server sync those signatures to Slave Servers.

## Slave Server

Slave Server provides signature update/download service to devices.



1. Architecture

# Action Flow

This section describes the action flow.

## Account Management



1. Account management action flow

1. addnew: add a new account;

2. delete: delete an account;

3. update: change password or email address.

## Device Type / Service Type Management

We adopt XML configuration files maintain Device Type / Service Type.

<DeviceTypes>

<DeviceType>UTM 100</DeviceType>

<DeviceType>UTM 200</DeviceType>

<DeviceType>UTM 300</DeviceType>

</DeviceTypes>

<ServiceTypes>

<ServiceType>AV</ServiceType>

<ServiceType>IDP</ServiceType>

</ServiceTypes>

## File management

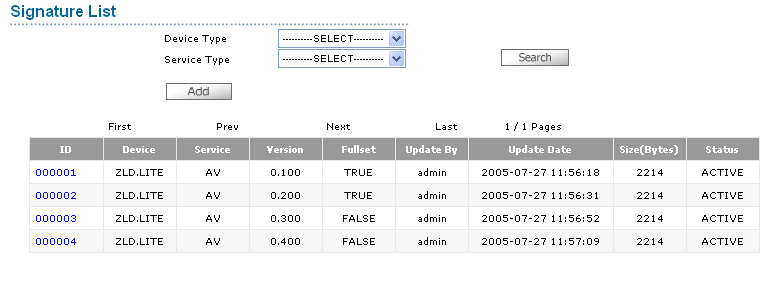


1. File management action flow
2. delete a file;
3. add a file;
4. update a file;
5. log the change of file;
6. synchronize with all these slavers;
7. if sync fail, fetch mail list from database;
8. if sync fail, send email to notify.

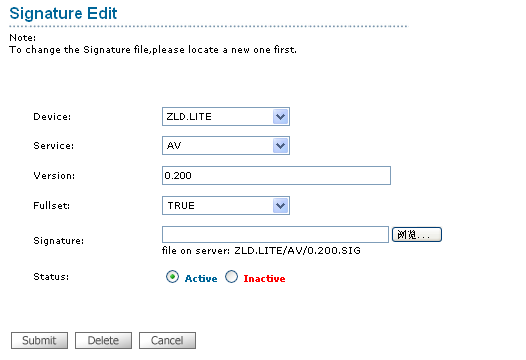
# GUI Layer

This section describes the GUI layer of Update Server. It is to be discussed, below is just the reference GUI.

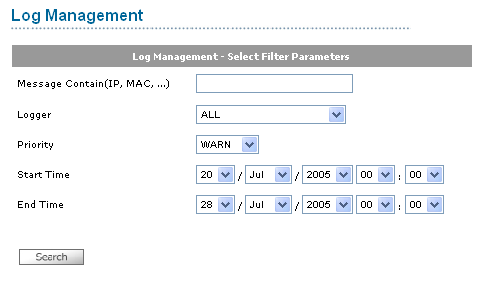
## Signature List

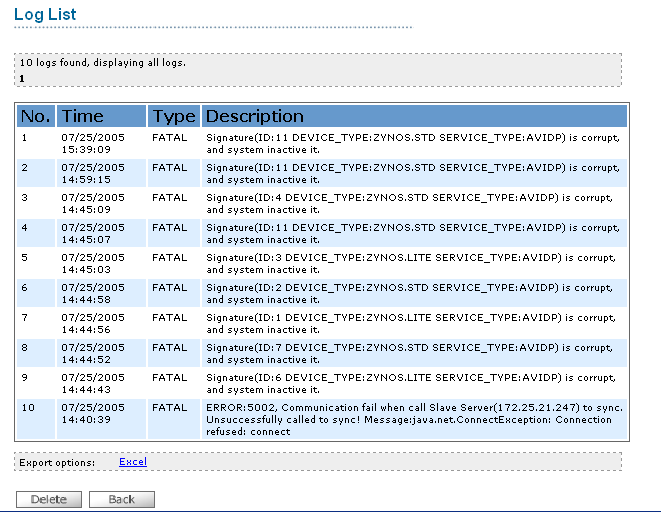


## Add / Modify Panel



## Log Management



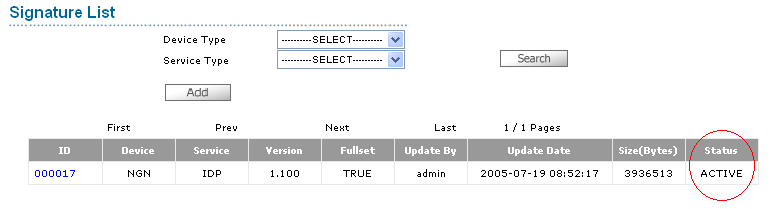


## Slave Server List

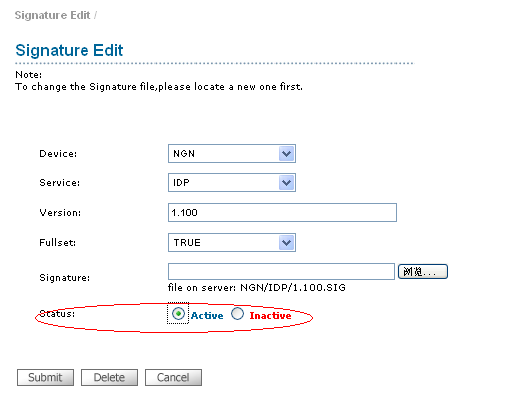


## GUI Modification

1. Signature
2. Signature list



1. Signature edit



## Signature File Name

Full signature file name extension: .sig

Incremental signature file name extension: .inc

When signature files are uploaded to Update Server, server saves then with extension .sig all.

# Signature Update / Download

## Background

The current design of Update Server does not prevent attack via http protocol. If a malicious attacker obtains the HTTP URL and repeatedly sends the download request, Update Server resource will be exhausted.

For security purpose, we will change the protocol to HTTPS and add authentication at Update Server.

## Flow

2. generate token

5. validate token

1. deviceType, serviceType, engineVer, sigVer, mac

3. code, message , filelist, version

4. token, mac

6. download signature

Update Server

https

https

Device

https

https

Get Token

Download

1. Signature Update Flow

Description:

1. Device send request to Update Server, parameters needed are deviceType, serviceType, engineVer, sigVer, mac
2. Update Server check signature update from database, generate token for signature file
3. Update Server send response to device, the information includes code, message, filelist, version
4. Device send download request to Update Server, parameters needed is token, mac
5. Update Server check whether the token is expired, then send the signature file
6. Device download signature
7. Repeats 4~6 if more than one token is returned in 3.

## API

***1.Get Token***

|  |  |  |  |
| --- | --- | --- | --- |
| Get token | | | |
| *Signature file update request from device.* | | | |
| **Request** | /upd/update?action=*sigUpdate*&deviceType=xxxx*&*  serviceType=xxx&engineVer=x.xx&sigVer=x.xx&mac=xxxxxxxxxxxx | | |
|  | Parameter | Value | Description |
| action | sigUpdate | System field |
| deviceType | Xxxx | Platform type, for example UTM |
| serviceType | Xxxx | Service type, for example AV,IDP |
| engineVer | x.xxx | engine version in device, for example: 2.000 |
| sigVer | x.xxx | Signature file version, for example: 2.340 |
| mac | Xxxxxxxxxxxxxxxxxx | Mac address |
| **Response** | *<returns>*  *<code>0</code>*  *<message>N/A</message>*  *<filelist>fileURL1#fileURL2</filelist>*  *<version>version1#version2</version>*  *</returns>* | | |
|  | Return | Value | Description |
| code | 0 | Successful |
| 1 | No update |
| <error code> | Fail; error code |
| message | N/A | Successful |
| Xxxxxxxxxxx | Fail; error message |
| filelist | https://update.titan.com/upd/dl?token=token1&mac=000000AAAAAA #https://update.titan.com/upd/dl?token=token2&mac=000000AAAAAA | Separate with “#” if there are more than one items |
| N/A |  |
| version | 1.330#1.340 | Success |
| N/A | Fail; |
| **Error Code** | Code | Message | |
| 1 | Current version x.xxx on device is latest. | |
| 9001 | Invalid or not supported model. | |
| 9002 | Invalid or not supported service. | |
| 9005 | Invalid or not supported signature version. | |
|  |  | |
| **Example** | Request | https://update.titan.com /upd/update  ?action=sigUpdate&deviceType=UTM&serviceType=idp&engineVer=1.200&sigVer=1.200&mac=000000AAAAAA | |
| Response | *<returns>*  *<code>0</code>*  *<message>N/A</message>*  *<filelist>* https://update.titan.com/upd/dl?token=token1&mac=000000AAAAAA # https://update.titan.com/upd/dl?token=token2&mac=000000AAAAAA *</filelist>*  *<version>*1.330#1.340*</version>*  *</returns>* | |

***2.Download***

|  |  |  |  |
| --- | --- | --- | --- |
| Download | | | |
| *Signature file download request from device.* | | | |
| **Request** | /upd/dl? token=token1&mac=xxxxxxxx | | |
|  | Parameter | Value | Description |
| token | xxxxxxxxxxxxxxxxxxxxxxxxxx | token |
| **Response** | *#case 1: success*  *Signature file*  *#case 2: token1 is invalid*  *<returns>*  *<code>9000</code>*  *<message>*Invalid or expired token.*</message>*  *</returns>* | | |
| **Error Code** | Code | Message | |
|  | 9000 | Invalid or expired token. | |
| **Example** | Request | https://update.titan.com/upd/dl?token=token1&mac=000000AAAAAA | |
| Response | Signature file | |

## Full-Incremental

As original design, any signatures update are full-set ones, the file size of full-set signatures is much

larger than incremental ones. Downloading full-set signatures cost too much network bandwidth. For the purpose to decrease size of signatures, we classify signatures into full and incremental ones. Device download incremental or full signatures. Incremental signature is compared to previous Full version. For example, V1.003 is compared to V1.000. If the Full version on device are the same as Server & the Incremental version on Server is newer than on device, device only download the last Incremental signature, delete the previous Incremental signature.

## Example

Case1:

**Last signature version on server is 1.003,**

**device signature is 1.000**

**Download (Last Incremental )**

**Download (1.003)**

Case2:

**Last signature version on server is 2.002,**

**device signature is 1.001**

**Download (Last Full, Last Incremental)**

**Download(2.000, 2.002)**

Case3:

**Last signature version on server is 3.000,**

**device signature is 2.003**

**Download (Last Full, Last Incremental)**

**Download(3.000, NULL)**

|  |  |
| --- | --- |
| Version | Full / Incremental |
| 1.000 | F |
| 1.001 | I |
| 1.002 | I |
| 1.003 | I |
| 2.000 | F |
| 2.001 | I |
| 2.002 | I |
| 2.003 | I |
| 3.000 | F |

## Auto Upgrade Schedule Arrangement

If we implement an auto upgrade mechanism in devices, please set the default value randomly. If we set a same default value on each device, then many devices may download signatures at that time point. So, please do set the default value randomly.

# Firmware Update / Download

## Flow

2. generate token

5. validate token

1. deviceType, fwVer, mac

3. code, message , filelist, version

4. token, mac

6. download

Update Server

https

https

Device

https

https

Get Token

Download

1. Firmware Update Flow

Description:

1. Device send request to Update Server, parameters needed are deviceType, fwVer, mac
2. Update Server check firmware update from database, generate token for firmware file
3. Update Server send response to device, the information includes code, message, filelist, version
4. Device send download request to Update Server, parameters needed is token, mac
5. Update Server check whether the token is expired, then send the signature file
6. Device download signature
7. Repeats 4~6 if more than one token is returned in 3.

## API

***1.Get Token***

|  |  |  |  |
| --- | --- | --- | --- |
| Get token | | | |
| *Firmware file update request from device.* | | | |
| **Request** | /upd/update?action=*fwUpdate*&deviceType=xxxx&fwVer=X.Y.Z-I&mac=xxxxxxxxxxxx | | |
|  | Parameter | Value | Description |
| action | fwUpdate | System field |
| deviceType | Xxxx | Platform type, for example UTM |
| fwVer | X.Y.Z-I | Firmware file version, for example: 2.3.4-5 |
| mac | Xxxxxxxxxxxxxxxxxx | Mac address |
| **Response** | *<returns>*  *<code>0</code>*  *<message>N/A</message>*  *<filelist>fileURL1 </filelist>*  *<version>version1#version2</version>*  *</returns>* | | |
|  | Return | Value | Description |
| code | 0 | Successful |
| 1 | No update |
| <error code> | Fail; error code |
| message | N/A | Successful |
| Xxxxxxxxxxx | Fail; error message |
| filelist | https://update.titan.com/upd/dl?token=token1&mac=000000AAAAAA | Separate with “#” if there are more than one items |
| N/A |  |
| version | 2.3.4-5 | Success |
| N/A | Fail; |
| **Error Code** | Code | Message | |
| 1 | Current version x.xxx on device is latest. | |
| 9001 | Invalid or not supported model. | |
| 9007 | Invalid or not supported firmware version. | |
|  |  | |
| **Example** | Request | https://update.titan.com /upd/update  ?action=fwUpdate&deviceType=UTM&fwVer=2.3.3&mac=000000AAAAAA | |
| Response | *<returns>*  *<code>0</code>*  *<message>N/A</message>*  *<filelist>* https://update.titan.com/upd/dl?token=token1&mac=000000AAAAAA # https://update.titan.com/upd/dl?token=token2&mac=000000AAAAAA *</filelist>*  *<version>*2.3.4-5*</version>*  *</returns>* | |

***2.Download***

|  |  |  |  |
| --- | --- | --- | --- |
| Download | | | |
| *Firmware file download request from device.* | | | |
| **Request** | /upd/dl? token=token1&mac=xxxxxxxx | | |
|  | Parameter | Value | Description |
| token | xxxxxxxxxxxxxxxxxxxxxxxxxx | token |
| **Response** | *#case 1: success*  *Firmware file*  *#case 2: token1 is invalid*  *<returns>*  *<code>9000</code>*  *<message>*Invalid or expired token.*</message>*  *</returns>* | | |
| **Error Code** | Code | Message | |
|  | 9000 | Invalid or expired token. | |
| **Example** | Request | https://update.titan.com/upd/dl?token=token1&mac=000000AAAAAA | |
| Response | Firmware file | |

## Firmware Version Format

**X.Y.Z-I**

**X: Production Release**

**Y: Check Point Release**

**Z: Periodical Release**

**I: Daily Build Release**

For official release, the format is Z.Y.Z.

X, Y, Z, I are all integer.

Version Comparison

We have two versions:

Version 1 (V1): X1.Y1.Z1-I1

Version 2 (V2): X2.Y2.Z2-I2



For two versions:

Version 1 (V1): X1.Y1.Z1

Version 2 (V2): X1.Y1.Z1-I1

V1>V2

## Auto Upgrade Schedule Arrangement

If we implement an auto upgrade mechanism in devices, please set the default value randomly. If we set a same default value on each device, then many devices may download firmwares at that time point. So, please do set the default value randomly.

# Token Mechanism

## Generate Token

Basic char set: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

Token = <Random String> +”;”+ <File Id>

Random String is a string combined with 10 chars, and each char is randomly got from basic char set above. There are 6210 combinations.

File Id is the unique id of each file.

## Token- file Mapping

Update Server maintain a mapping of token-file in memory. Each token is mapping to a file. And each token can exists in memory for 5 minutes. Once the token is expired, it is removed from memory.

## Download

Once a download is completed, the related token is removed from memory.