# Installation Guide

# FreeRadius Authy Multifactor Authentication

# Table of Contents

**1 DOCUMENT CONTROL ..... 4**
**2 INTRODUCTION ..... 5**
2.1 PURPOSE ..... 5
2.2 SCOPE ..... 5
2.3 INTENDED AUDIENCE ..... 5
**3 OVERVIEW ..... 5**
3.1 AUTHY MFA MODULES ..... 5
*3.1.1 Mapper Module ..... **Error! Bookmark not defined.***
*3.1.2 Authy MFA Module ..... **Error! Bookmark not defined.***
3.2 LOGGING ..... 5
**4 PREREQUISITES ..... 5**
4.1 SOFTWARE ..... 5
4.2 OPERATING SYSTEM PACKAGES ..... 6
*4.2.1 Red Hat Enterprise Linux ..... 6*
**5 REFERENCE TABLE ..... 7**
**6 INSTALLATION ..... 8**
6.1 DEPLOYING THE AUTHY MFA MODULES ..... 8
6.2 CONFIGURING FREERADIUS ..... 8
**7 CALLBACK SERVER (RECOMMENDED) ..... 12**
**8 MODULE BEHAVIOR CONFIGURATION FILE ..... 14**
8.1 CONFIGURATION TEMPLATE ..... 14
8.2 CONFIGURATION DETAILS ..... 16
*8.2.1 RADIUS ..... 16*
*8.2.2 Auth ..... 17*
*8.2.3 OTP ..... 17*
*8.2.4 OneTouch ..... 18*
*8.2.5 IDStore ..... 20*
8.3 SAMPLE CONFIGURATIONS ..... 21
*8.3.1 TOTP with Challenge Response ..... 21*
*8.3.2 TOTP with no Challenge Response ..... 22*
*8.3.3 OneTouch Authentication ..... 22*
*8.3.4 OneTouch and TOTP Authentication ..... 23*
8.4 INCOMPATIBLE CONFIGURATIONS ..... 24
**9 MESSAGE CONFIGURATION FILE ..... 24**
9.1 MESSAGES ..... 24
*9.1.1 Logging Messages ..... 24*
*9.1.2 End User Facing Messages ..... 28*
9.2 CONFIGURATION ERROR LOG MESSAGES ..... 29
9.3 SAMPLE MESSAGE FILE ..... 31
**10 ADDITIONAL USAGE NOTES ..... 34**

10.1 ONE CONFIGURATION "STYLE" PER FREERADIUS INSTALLATION ..... 34

# 1 Document Control

This section details the document version history, along with reviews and approvals performed per version.

**Review and Approval**

| Name | Signature | Project Role | Version | Review Date |
|---|---|---|---|---|
| | | | | |

**Revision History**

| Version | Issue Date | Description of Version/Changes | Author |
|---|---|---|---|
| 0.1 | 12/07/2016 | Initial draft | Toshie Takahashi |

# 2 Introduction

## 2.1 Purpose

This guide will provide a step-by-step introduction to utilizing Authy's Time-based One Time Password (TOTP) and OneTouch features in a FreeRadius environment. The primary function of the features mentioned in this document is for the use with OpenVPN and Cisco AnyConnect Virtual Private Network (VPN) servers that will utilize FreeRadius for backend authentication. This document assumes the working environment is Linux based.

## 2.2 Scope

This document is not intended for the purposes of the installing or configuration of FreeRadius or any VPN servers or clients. Any configuration changes and prerequisites required to implement the MFA features will be listed.

## 2.3 Intended Audience

This document is written for administrators to implement a multifactor authentication solution to FreeRadius using Authy.

# 3 Overview

## 3.1 Authy MFA Module

The Authy MFA module is a perl script designed to work with FreeRadius's perl module, rlm_perl. The MFA module will handle requests and communicate to Authy for TOTP and OneTouch based requests. The behavior can differ based on configuration. This module will validate TOTP tokens against Authy, as well as poll the target server at set intervals to check on OneTouch request status.

## 3.2 Logging

All of the modules defined in this document will output logging to the FreeRadius logs.

# 4 Prerequisites

The following are prerequisites in order to utilize the MFA modules for FreeRadius.

## 4.1 Software

| Product | Version | Description |
|---|---|---|
| FreeRadius | 3.0.12 | The RADIUS server used for backend authentication. This will execute the MFA modules. The perl module is expected to be available from the initial install of FreeRadius. |

| | | |
|---|---|---|
| | | Refer to the FreeRadius installation guide to complete this prerequisite at http://wiki.freeradius.org/building/Home |
| Perl | 5.10.1 | The programming language the modules are written in. |
| Red Hat Enterprise Linux | 6.5 | The operating system version these modules have been created on. |

## 4.2 Operating System Packages

### 4.2.1 Red Hat Enterprise Linux

Many of these libraries should be installed if there is an existing FreeRadius installation. Some of these libraries are necessary for code specific to the Authy MFA modules.

| Package Name | Description |
|---|---|
| pcre | Perl compatible regular expressions. |
| pcre-devel | Perl compatible regular expressions |
| libtalloc | Used for memory allocation |
| libtalloc-devel | Used for memory allocation |
| httpd-devel | Development interface for HTTP server |
| openssl | Toolkit allowing the use of TLS and SSL |
| openssl-devel | Toolkit allowing the use of TLS and SSL |
| perl | Allows the use of perl |
| perl-devel | Allows the use of perl |
| openldap | Allows the use of OpenLDAP libraries |
| openldap-devel | Allows the use of OpenLDAP libraries |
| curl | Toolkit allowing data transfer through URLs |
| curl-devel | Toolkit allowing data transfer through URLs |
| perl-LDAP | Allows the use of LDAP calls in perl |
| Development Tools | Developer toolkit |

# 5 Reference Table

This document will use placeholders throughout the instructions as each environment may have different install paths or desired locations. The following table will outline what each placeholder is used for. The environment value column is intentionally left blank to fill in with environmentally specific variables by the user.

| Placeholder Name | Description | Environment Value |
|---|---|---|
| FreeRadius References | | |
| <FREERADIUS_HOME> | The location FreeRadius is installed | |
| <SITE_NAME> | The FreeRadius site that will have MFA functionality configured. Defaults to the **default** site. | |
| Tomcat References | | |
| <AUTHY_API_KEY> | The Authy API Key. Used by the callback application to validate the authenticity of callback and polling requests. | |
| <AUTHY_LOG_LOCATION> | The location to store callback logs. | |

# 6 Installation

## 6.1 Deploying the Authy MFA Modules

1. Ensure the FreeRadius server is shutdown.
2. Navigate to **mods-config/perl**.

```
cd <FREERADIUS_HOME>/etc/raddb/mods-config/perl
```

3. Move the perl scripts and configuration file to this directory via FTP or similar method. The directory structure should look like the following afterwards:

```
perl/
        authy-mapper.pl
        authy-password-parser.pl
        authy-mfa.pl
        config.ini
```

## 6.2 Configuring FreeRadius

1. Ensure the FreeRadius server is shutdown.
2. Navigates to **mods-available/perl**.

```
cd <FREERADIUS_HOME>/etc/raddb/mods-available/
```

3. Make a backup of the perl configuration file.

```
cp perl perl.ORIG
```

**Note**: Ensure mods-available/perl exists. This module is available as part of the FreeRadius installation.

4. Navigate to **mods-enabled/perl**.

```
cd <FREERADIUS_HOME>/etc/raddb/mods-enabled/
```

5. Create a symbolic link to <FREERADIUS_HOME>/mods-available/perl if it does not exist already.

ln –s ../mods-available/perl perl

6. Edit the perl file.

```
vim perl
```

7. Add the following contents to the file.

```
perl authy-mapper {
        perl_flags=-I${modconfdir}/perl
        filename=${modconfdir}/perl/authy-mapper.pl
        func_authorize = authorize
}

perl authy-password-parser {
        perl_flags=-I${modconfdir}/perl
        filename=${modconfdir}/perl/authy-password-parser.pl
        func_authorize = authorize
}

perl authy-mfa {
        perl_flags=-I${modconfdir}/perl
        filename=${modconfdir}/perl/authy-mfa.pl
        func_authorize = authorize
        func_authenticate = authenticate
}
```

8. Save and exit the file.
9. Navigate to <FREERADIUS_HOME>/sites-available

```
cd <FREERADIUS_HOME>/etc/raddb/sites-available/<SITE_NAME>
```

10. Make a backup of the <SITE_NAME> file.

```
cp <SITE_NAME> <SITE_NAME>.ORIG
```

11. Navigate to <FREERADIUS_HOME>/sites-enabled

```
cd <FREERADIUS_HOME>/sites-enabled/
```

12. Create a symbolic link to <FREERADIUS_HOME>/sites-enabled/<SITE_NAME> if it does not exist already.

```
ln –s ../sites-enabled/<SITE_NAME> <SITE_NAME>
```

13. Edit the <SITE_NAME> file.

```
vim <SITE_NAME>
```

14. Add the following contents to the existing **authorize** block.

```
authorize {
        …
        authy-mapper
        authy-password-parser
        authy-mfa
        …
}
```

15. Add the following contents to the existing **authenticate** block.

```
authenticate {
        …
        Auth-Type authy {
                authy-mfa
```

```
        }

        Auth-Type premfa {
                …
                authy-mfa
        }
}
```

**Note**: In the **Auth-Type premfa** block, insert any modules that should be executed prior to MFA such as an LDAP authentication module. The module should be placed above the **authy-mfa** module. For example, the premfa block could look like the following:

```
Authy-Type premfa{
        ldap
        authy-mfa
}
```

16. Save and exit the file.
17. Start the FreeRadius server.

# 7 Callback Server (Recommended)

Authy provides two ways to validate a OneTouch request status. Polling directly to the Authy servers or providing Authy with a callback URL which will be updated when the request status changes. Utilizing the callback functionality will see improved performance as the size of the userbase scales. The Authy FreeRadius module cannot function as a callback server and is dependent on another entity to store the status for the module.

These instructions will describe the steps required to setup the callback server.

The following are prerequisites in order to utilize the MFA modules for FreeRadius.

## 7.1 Software

| Product | Version | Description |
|---|---|---|
| Apache Tomcat | 8.0.39 | The application server that will host the callback application. Configuration and basic security considerations can be found at https://tomcat.apache.org/tomcat-8.0-doc/index.html |
| Java | 1.7.0_121 | The programming language the callback application utilizes. |

## 7.2 Network

The machine that will run the callback server will need to be accessible from the internet as Authy will need to call the application. It is not recommended to expose an application server such as Tomcat to the internet directly. An alternative is to have an HTTP proxy for the application server that will filter and remove bad or malicious requests.

## 7.3 Deployment

1. Create a setenv.sh file in Tomcat if it does not exist already

```
touch <TOMCAT_HOME>/bin/setenv.sh
```

2. Add the following lines to the file.

```
export AUTHY_API_KEY=<AUTHY_API_KEY>
```

```
export AUTHY_LOG_LOCATION=<AUTHY_LOG_LOCATION>
```

3. Start the Tomcat server.
4. Access the Tomcat server's deployment console in the browser.
5. Upload the AuthyCallback WAR file to the tomcat server and deploy.

**Note**: Alternatively the deployment can be done via command line by copying the WAR file to the tomcat WEBAPPS directory, then restarting the server.

6. Login to the Authy dashboard for the application that will utilize the multifactor authentication flow.
7. Set the OneTouch callback URL to https://<CALLBACK_HOST>:<CALLBACK_PORT>/AuthyCallback/callback and use a GET method.
8. Configure the **CustomPollingEndpoint** value in the Authy MFA module's configuration file to point to the callback host.

# 8 Module Behavior Configuration File

Configuration options for the Authy MFA modules will be defined in a separate file. The configuration options will be described below along with valid values.

## 8.1 Configuration Template

| RADIUS | | |
|---|---|---|
| Configuration Name | Accepted Values | Environment Value |
| IDParam | Any string value | |
| OTPParam | Any string value | |
| ReplyAuthType | Any string value | |
| StateMarker | Any string value | |

| Auth | | |
|---|---|---|
| Configuration Name | Accepted Values | Environment Value |
| Interactive | yes/no | |
| MaxAttempts | Any positive integer value | |
| OTPOption | Any string value | |
| OneTouchOption | Any string value | |

| OTP | | |
|---|---|---|
| Configuration Name | Accepted Values | Environment Value |
| Enabled | yes/no | |
| Delimiter | Any string value | |
| UseSandboxAPI | yes/no | |
| AlwaysSendSMS | yes/no | |
| VerifyUnregisteredUsers | yes/no | |

| OneTouch | | |
|---|---|---|
| Configuration Name | Accepted Values | Environment Value |
| Enabled | yes/no | |
| UseSandboxAPI | yes/no | |
| CustomPollingEndpoint | Any URL string value | |
| PollingInterval | Any decimal value | |
| ApprovalRequestTimeout | Any integer value | |

| | | |
|---|---|---|
| ApprovalRequestMessage | Any string value | |
| DefaultLogoURL | Any URL string value | |
| LowResLogoURL | Any URL string value | |
| MedResLogoURL | Any URL string value | |
| HighResLogoURL | Any URL string value | |

| IDStore (LDAP Mapper) | | |
|---|---|---|
| Configuration Name | Accepted Values | Environment Value |
| URI | Any LDAP string value | |
| CA | Any string value | |
| BindDN | Any string value | |
| UserBaseDN | Any string value | |
| SearchAttribute | Any string value | |
| IDAttribute | Any string value | |

| IDStore (Flatfile Mapper) | | |
|---|---|---|
| Configuration Name | Accepted Values | Environment Value |
| | | |

## 8.2 Configuration Details

This section will describe the functions of each configurable parameter in the configuration file. Many of these should remain as provided unless a known conflict exists in the FreeRadius environment. If a change of a value is required - or should be under consideration for change – it will be specified in the **Change Required** column. The possible values in the column are as follows:

**Yes** indicates a change to the configuration file should be changed to a value specific to the environment.
**No** indicates a change should not be made unless absolutely required.
**ENV** indicates the environment or business policies should be analyzed for the proper value and a default value may not be provided or sufficient.

Some configurations will have a default value if not set, indicated in the **description** field. If a configuration that has no default value is not set in the configuration file, the modules will error and stop.

### 8.2.1 RADIUS

| Configuration Name | Description | Change Required |
|---|---|---|
| IDParam | The key used to store the Authy ID parameter within the FreeRadius request. This value should only be changed if there is a known key conflict with other FreeRadius modules.<br><br>Ensure the value specified in this configuration exists in the FreeRadius **dictionary** file as an attribute.<br><br>**Default Value**: Authy-ID | No |
| OTPParam | The key used to store the OTP token parameter within the FreeRadius request. This value should only be changed if there is a known key conflict with other FreeRadius modules.<br><br>Ensure the value specified in this configuration exists in the FreeRadius **dictionary** file as an attribute.<br><br>**Default Value**: Authy-OTP | No |
| ReplyAuthType | The name for the authentication type that the custom modules will use. This value should only be changed if there is a known conflict with an existing Auth-Type in FreeRadius. | No |

| | | |
|---|---|---|
| | **Default Value**: authy-reply | |
| StateMarker | The prefix used to maintain state across FreeRadius and client challenge responses. The modules will analyze state to determine if it should handle a FreeRadius request. The Authy modules will only handle requests incoming with no state, or state prefixed with this value. This value should only be changed if there is a known conflict with other states maintained by FreeRadius.<br><br>**Default Value**: HCM::AuthyMFAState | No |

### 8.2.2 Auth

| Configuration Name | Description | Change Required |
|---|---|---|
| Interactive | **yes** if the client supports challenge responses.<br>**no** if the client does not support challenge responses.<br><br>**Default Value**: no | ENV |
| MaxAttempts | The maximum number of OTP attempts should be made before the client is responded with a REJECT response.<br><br>**Default Value**: 1 | ENV |
| OTPOption | The string value indicating the user has selected the OTP option. This value is only necessary if both OTP and OneTouch features are simultaneously enabled. This value should not be the same as **OneTouchOption**. | ENV |
| OneTouchOption | The string value indicating the user has selected the OneTouch option. This value is only necessary if both OTP and OneTouch features are simultaneously enabled. This value should not be the same as **OTPOption**. | ENV |

### 8.2.3 OTP

| Configuration Name | Description | Change Required |
|---|---|---|
| Enabled | **yes** if OTP validation is the desired flow for multifactor authentication.<br>**no** if OTP validation is not desired.<br><br>If both **OTP** and **OneTouch** are enabled, an extra challenge response will be sent to the client | ENV |

| | | |
|---|---|---|
| | prompting the user to select which authentication method is desired.<br><br>If OTP is disabled, then all configurations in this section will be ignored.<br><br>**Default Value**: no | |
| Delimiter | This configuration is only used **Interactive** in the **Auth** section is set to **no**. This value will determine the delimiter string used to separate a password value from the OTP in the case the OTP will be provided in a <password><delimiter><OTP> format. | ENV |
| UseSandboxAPI | **yes** if the Authy sandbox API endpoint will be used. An appropriate API key for the sandbox environment should also be set as the **AUTHY_SANDBOX_API_KEY** environmental variable.<br><br>**no** if the Authy production API endpoint will be used. An appropriate API key for the production environment should also be set as the **AUTHY_PROD_API_KEY** environmental variable.<br><br>**Default Value**: no | ENV |
| AlwaysSendSMS | **yes** to send the OTP token via SMS<br>**no** to send only a push notification if the user's phone is a smartphone with Authy installed.<br><br>**Default Value**: no | ENV |
| VerifyUnregisteredUsers | <br><br>**Default Value**: no | TBD |

### 8.2.4 OneTouch

| Configuration Name | Description | Change Required |
|---|---|---|
| Enabled | **yes** if OneTouch validation is the desired flow for multifactor authentication.<br>**no** if OneTouch validation is not desired. | ENV |

| | | |
|---|---|---|
| | If both **OTP** and **OneTouch** are enabled, an extra challenge response will be sent to the client prompting the user to select which authentication method is desired.<br><br>If OneTouch is disabled, then all configurations in this section will be ignored.<br><br>**Default Value**: no | |
| UseSandboxAPI | **yes** if the Authy sandbox API endpoint will be used. An appropriate API key for the sandbox environment should also be set as the **AUTHY_SANDBOX_API_KEY** environmental variable.<br><br>**no** if the Authy production API endpoint will be used. An appropriate API key for the production environment should also be set as the **AUTHY_PROD_API_KEY** environmental variable.<br><br>**Default Value**: no | ENV |
| CustomPollingEndpoint | The URL for the callback server setup to handle Authy OneTouch callbacks. If this configuration is not set, the module will communicate directly with Authy to determine the status of OneTouch requests.<br><br>Configuration of a callback server is recommended if performance is a concern. | Yes |
| PollingInterval | The interval in seconds to wait between polling requests for OneTouch status.<br><br>**Default Value**: 0.5 | ENV |
| ApprovalRequestTimeout | The expiration time to set for OneTouch approval requests.<br><br>**Default Value**: 86400 | ENV |
| ApprovalRequestMessage | The message to send to the User along with the OneTouch request. | Yes |
| DefaultLogoURL | The URL to the default logo to display to users in the OneTouch request.<br><br>If this configuration is not set, no custom image will be displayed to the user. | ENV |

| | | |
|---|---|---|
| | If any of **LowResLogoURL**, **MedResLogoURL**, or **HighResLogoURL** are set, this configuration must also be set. | |
| LowResLogoURL | The URL to the low-resolution logo to display to users in the OneTouch request.<br><br>If this configuration is not set, no custom low-resolution image will be displayed to the user. | ENV |
| MedResLogoURL | The URL to the normal resolution logo to display to users in the OneTouch request.<br><br>If this configuration is not set, no custom normal resolution image will be displayed to the user. | ENV |
| HighResLogoURL | The URL to the high-resolution logo to display to users in the OneTouch request.<br><br>If this configuration is not set, no custom high-resolution image will be displayed to the user. | ENV |

### 8.2.5 IDStore

#### 8.2.5.1 LDAP Mapper

| Configuration Name | Description | Change Required |
|---|---|---|
| URI | The LDAP URI. This value should be of the form **ldap(s)://<host>:<port>**. The use of LDAPS is strongly recommended.<br><br>**Example**: ldaps://example.com:636 | Yes |
| CA | The location to find the certificate store to connect securely to LDAP.<br><br>**Example**: /tmp/ca | Yes |
| BindDN | The account to connect to LDAP with to retrieve the AuthyID attribute value. The use of a service account instead of an administrative account is strongly recommended.<br><br>**Example**: cn=authysvc,dc=example,dc=com | Yes |
| UserBaseDN | The most specific DN containing all the users that should be able to utilize Authy multifactor authentication. | Yes |

| | | |
|---|---|---|
| | **Example**: ou=Users,dc=example,dc=com | |
| SearchAttribute | The attribute used to find a user in LDAP. In some directories this will value will be **uid**. In Active Directory environments this is usually **sAMAccountName**.<br><br>**Example**: uid | Yes |
| IDAttribute | The user attribute in LDAP storing the AuthyID attribute.<br><br>**Example**: authyId | Yes |

*The bind password for the **BindDN** value should be stored in an environmental value to retrieve.

#### 8.2.5.2 Flatfile Mapper

| Configuration Name | Description | Change Required |
|---|---|---|
| | | |

## 8.3 Sample Configurations

This section will provide sample configurations for sample use cases.

### 8.3.1 TOTP with Challenge Response

This configuration would be used if the client supports challenge responses and the TOTP verification is the only method desired. This configuration also utilizes the LDAP Mapper.

```
[RADIUS]
IDParam = Authy-ID
OTPParam = Authy-OTP
ReplyAuthType = authy-reply
StateMarker = HCM::AuthyMFAState

[Auth]
Interactive = yes
MaxAttempts = 3

[OTP]
Enabled = yes
UseSandboxAPI = no
```

```
AlwaysSendSMS = no
VerifyUnregisteredUsers = no

[OneTouch]

[IDStore]
URI = ldaps://example.com:636
CA = /tmp/ca
BindDN = cn=Directory Manager
UserBaseDN = dc=example,dc=com
SearchAttribute = uid
IDAttribute = authyId
```

### 8.3.2 TOTP with no Challenge Response

This configuration would be used if the client does not support challenge responses and the TOTP verification is the only method desired. This configuration also utilizes the LDAP Mapper.

```
[RADIUS]
IDParam = Authy-ID
OTPParam = Authy-OTP
ReplyAuthType = authy-reply
StateMarker = HCM::AuthyMFAState

[Auth]
Interactive = no

[OTP]

[OneTouch]
Enabled = yes


[IDStore]
URI = ldaps://example.com:636
CA = /tmp/ca
BindDN = cn=Directory Manager
UserBaseDN = dc=example,dc=com
SearchAttribute = uid
IDAttribute = authyId
```

### 8.3.3 OneTouch Authentication

This configuration would be used if OneTouch verification is the only method desired. This configuration also utilizes the LDAP Mapper.

```
[RADIUS]
```

```
IDParam = Authy-ID
OTPParam = Authy-OTP
ReplyAuthType = authy-reply
StateMarker = HCM::AuthyMFAState

[Auth]
Interactive = no

[OTP]

[OneTouch]
Enabled = yes
UseSandboxAPI = no
CustomPollingEndpoint = https://callback.example.com/polling
PollingInterval = 0.5
ApprovalRequestTimeout = 86400
ApprovalRequestMessage = FreeRadius has requested your approval.

[IDStore]
URI = ldaps://example.com:636
CA = /tmp/ca
BindDN = cn=Directory Manager
UserBaseDN = dc=example,dc=com
SearchAttribute = uid
IDAttribute = authyId
```

### 8.3.4 OneTouch and TOTP Authentication

This configuration would be used if the user should decide if OneTouch or TOTP verification should be used. The client **MUST** support challenge responses to use this configuration. This configuration also uses the LDAP Mapper.

```
[RADIUS]
IDParam = Authy-ID
OTPParam = Authy-OTP
ReplyAuthType = authy-reply
StateMarker = HCM::AuthyMFAState

[Auth]
Interactive = yes
MaxAttempts = 5
OTPOption = 1
OneTouchOption = 2

[OTP]
Enabled = yes
```

```
UseSandboxAPI = no
AlwaysSendSMS = no
VerifyUnregisteredUsers = no

[OneTouch]
Enabled = yes
UseSandboxAPI = no
CustomPollingEndpoint = https://callback.example.com/polling
PollingInterval = 0.5
ApprovalRequestTimeout = 86400
ApprovalRequestMessage = FreeRadius has requested your approval.

[IDStore]
URI = ldaps://example.com:636
CA = /tmp/ca
BindDN = cn=Directory Manager
UserBaseDN = dc=example,dc=com
SearchAttribute = uid
IDAttribute = authyId
```

### 8.4 Incompatible Configurations

There are numerous configurations that may conflict with each other or may not work in a given environment. This section will outline some of these conflicting configurations.

1. Both OTP and OneTouch are disabled.
2. Both OTP and OneTouch are enabled but the client does not support challenge responses.
3. The Password Parser is disabled and the client does not support challenge responses.
4. Modules outside the scope of this document already utilize the AuthyIDParam or AuthyTokenParam values.
5. Any required configuration is not set in the configuration file.

## 9 Message Configuration File

A message configuration file is also specified in the case that messages to the end user via challenge response or messages in the log need to be customized.

### 9.1 Messages

#### 9.1.1 Logging Messages

| Message Name | Description |
|---|---|
| SplittingPassword | The log stating the delimiter used, and the action being taken on the password field. |

| | |
|---|---|
| | Log appears if client does not support challenge responses. |
| PlacingPasswordAndOTPIntoRequest | The log stating the password parsing has succeeded and is storing the values in the FreeRadius request object.<br><br>Log appears if client does not support challenge responses. |
| CheckingStateCompatibility | The log stating the module is analyzing the FreeRadius request state.<br><br>Log appears if the client either supports challenge responses or has no state. |
| UpdatingAuthTypeToReply | The log stating the module has found it is capable of handling the FreeRadius request in its current state.<br><br>Log appears if the client supports challenge responses or has no state. |
| PerformingSilentAuthn | The log stating the client does not support challenge responses.<br><br>Log appears if the client does not support challenge responses. |
| PerformingInteractiveAuthnWithoutState | The log stating the process will enter a flow providing challenge responses but the request state has not yet been modified.<br><br>Log appears if client supports challenge responses and the request process has just begin. |
| PerformingInteractiveAuthnWithState | The log stating the process has returned from a challenge response and the module will begin processing the response.<br><br>Log appears if the client is returning from a challenge response. |
| AskingForAuthnMethod | The log stating both OTP and OneTouch authentication methods are enabled and requires user input to determine flow.<br><br>Log appears if both OTP and OneTouch methods are enabled and the client supports challenge responses. |

| | |
|---|---|
| InvalidAuthnMethod | The log stating the user has provided an invalid authentication method.<br><br>Log appears if both OTP and OneTouch methods are enabled but the user has provided a response that maps to neither the **OTPOption** or **OneTouchOption** values. |
| MarkingFailedAuthnAttempt | The log stating the number of authentication attempts has been incremented.<br><br>Log appears if the OTP flow is used and an OTP validation failed. |
| UnexpectedAuthyResponse | The log stating an unexpected response from Authy has been found when validating OTP or when creating a OneTouch request.<br><br>Log appears if an unexpected Authy response occurs. |
| VerifyingOTP | The log stating verification of OTP against Authy is beginning.<br><br>Log appears if the user provided an OTP. |
| SendingOTPRequest | The log stating the OTP flow is used and the user is sent an OTP notification on the mobile device.<br><br>Log appears if the client supports challenge responses and the OTP flow is used. |
| ParsingOTPResponse | The log stating the user has provided the OTP to the challenge prompt.<br><br>Log appears if the client supports challenge responses and the OTP flow is used. |
| AskingForOTP | The log stating the user will be prompted for the OTP.<br><br>Log appears if the client supports challenge responses and the OTP flow is used. |
| ParsingOTPVerificationResponse | The log stating the interpretation of the challenge response has begun.<br><br>Log appears if the OTP flow is used. |
| OTPAccepted | The log stating Authy has deemed the OTP valid. |

| | |
|---|---|
| | Log appears if the OTP flow is used and Authy returned a successful response. |
| OTPRejected | The log stating Authy has deemed the OTP invalid.<br><br>Log appears if the OTP flow is used and Authy returned a failure response. |
| CreatingOneTouchApprovalRequest | The log stating a OneTouch approval request is being created by Authy.<br><br>Log appears if the OneTouch flow is used. |
| ParsingOneTouchApprovalRequestCreationResponse | The log stating a OneTouch approval request creation response is being parsed.<br><br>Log appears if the OneTouch flow is used and the OneTouch approval request has been submitted. |
| PollingOneTouchEndpoint | The log stating the module is polling the target endpoint to check on the OneTouch request status.<br><br>Log appears if the OneTouch status has not entered an ACCEPTED or DENIED state and the request has not expired. |
| OneTouchApprovalRequestApproved | The log stating the OneTouch status has entered an ACCEPTED state.<br><br>Log appears if OneTouch status has changed to ACCEPTED since the last polling interval. |
| OneTouchApprovalRequestDenied | The log stating the OneTouch status has entered a DENIED state.<br><br>Log appears if OneTouch status has changed to DENIED since the last polling interval. |
| OneTouchApprovalRequestExpired | The log stating the OneTouch status has passed the expiration period.<br><br>Log appears if OneTouch status has not changed after polling for the set **ApprovalRequestTimeout** period. |
| ValidatingAuthzRequest | The log stating the module is analyzing FreeRadius authorization request. |

| | |
|---|---|
| ValidatingAuthnRequest | The log stating the module is analyzing FreeRadius authentication request. |
| ValidatingOTP | The log stating the module is validating the OTP.<br><br>Log appears if the OTP flow is used. |
| ValidatingState | The log stating the module is validating the current FreeRadius request state.<br><br>Log appears if the client supports challenge responses and the OTP flow is used. |
| ValidatingAuthnMethodChoice | The log stating the user choice is being validated.<br><br>Log appears if the client supports challenge responses and both OTP and OneTouch flows are enabled. |

### 9.1.2 End User Facing Messages

| Message Name | Description |
|---|---|
| EnterAuthnMethod | Message sent to client when both OTP and OneTouch authentication methods are enabled. This message should communicate the **OTPOption** and **OneTouchOption** values to the user in some capacity. |
| ReenterAuthnMethod | Message sent to client when the user has provided an invalid authentication method. Ensure the user provides one of the values set in the **OTPOption** or **OneTouchOption** configuration. |
| EnterAuthnMethodAfterOTP | Message sent to client after the **MaxAttempts** threshold has been exceeded and both OTP and OneTouch authentication methods are supported. |
| EnterAuthnMethodAfterOneTouch | Message sent to client after a OneTouch **ApprovalRequestTimeout** expiration threshold has been exceeded and both OTP and OneTouch authentication methods are supported. |
| EnterOTP | Message sent to client when OTP flow is used and the client supports challenge responses. |

| | |
|---|---|
| ReenterOTP | Message sent to client when OTP flow is used and the client supports challenge responses but the OTP submission has been deemed invalid. This message will only appear if **MaxAttemtps** threshold has not been exceeded. |
| AuthnSucceeded | Message sent to client when the modules have successfully completed. |
| AuthnFailed | Message sent to client when the modules have determined authentication failure. |

## 9.2 Configuration Error Log Messages

| Message Name | Description |
|---|---|
| InvalidConfigInt | Error message logged when the configuration parser expected an integer but retrieved a different value. |
| InvalidConfigBool | Error message logged when the configuration parser expected a Boolean value but retrieved a different value. |
| IDAndOTPParamsConflict | Error message logged when the **IDParam** and **OTPParam** values in the configuration file are the same. These values must be unique to each other. |
| NoProductionAPIKey | Error message when no production API key is set in the environmental variable **AUTHY_PRODUCTION_API_KEY**.<br><br>Logged only if **UseSandboxAPI** is disabled. |
| NoSandboxAPIKey | Error message when no sandbox API key is set in the environmental variable **AUTHY_SANDBOX_API_KEY**.<br><br>Logged only if **UseSandboxAPI** is enabled. |
| InvalidProductionAPIKeyVerificationResponse | Error message when an unexpected response is received from Authy when using the production API key. |
| InvalidSandboxAPIKeyVerificationResponse | Error message when an unexpected response is received from Authy when using the sandbox API key. |
| ProductionAPIKeyVerificationFailed | Error message when Authy fails verification of production API key. |
| SandboxAPIKeyVerificationFailed | Error message when Authy fails verification |

| | |
|---|---|
| | of sandbox API key. |
| InvalidProductionAPIKey | Error message when Authy deems production API key is invalid. |
| InvalidSandboxAPIKey | Error message when Authy deems sandbox API key is invalid. |
| InvalidMaxAttemptCount | Error message when the **MaxAttempts** configuration is not a positive integer. |
| NoAuthnMethods | Error message when neither **OTP** nor **OneTouch** configurations are enabled. |
| NoOTPOption | Error message when both **OTP** and **OneTouch** configurations are enabled but **OTPOption** is not configured. |
| NoOneTouchOption | Error message when both **OTP** and **OneTouch** configurations are enabled but **OneTouchOption** is not configured. |
| OTPAndOneTouchOptionsConflict | Error message when both **OTP** and **OneTouch** configurations are enabled and **OTPOption** and **OneTouchOption** are set to the same value. **OTPOption** and **OneTouchOption** must be unique values to each other. |
| NoOTPDelimiter | Error message when **Interactive** is set to **no**, **OTP** is enabled, but the **Delimiter** configuration is not set. |
| InvalidOneTouchPollingInterval | Error message when the **PollingInterval** configuration is invalid. |
| InvalidOneTouchApprovalRequestTimeout | Error message when the **ApprovalRequestTimeout** configuration is less than 0. |
| NoOneTouchApprovalRequestMessage | Error message when **OneTouch** is enabled but no **ApprovalRequestMessage** is configured. |
| NoOneTouchDefaultLogoURL | Error message when **OneTouch** is enabled but no **DefaultLogoURL** is configured. |
| InvalidState | Error message when the **StateMarker** indicates the module can handle the request but the contents of the state are invalid. |
| UnexpectedOTPParam | Error message when client sends OTP but the configuration does not expect it. |
| NoUserNameInRequest | Error message when the request does not contain a username. |
| NoPasswordInRequest | Error message when the request does not contain a password |
| NoIDInRequest | Error message when no AuthyID is found in |

| | |
|---|---|
| | the request. |
| NoOTPInRequest | Error message when no OTP is found in the request but is expected. |
| NoChallengeResponseInRequest | Error message when the challenge response reply is empty. |
| InvalidOTP | Error message when the OTP token length is invalid. |
| OTPRequestFailed | Error message when the request to validate the OTP token has failed client side. |
| InvalidOTPResponse | Error message when the response to validate the OTP token is invalid. |
| OTPVerificationRequestFailed | Error message when Authy determines the OTP token validation request was invalid. |
| InvalidOTPVerificationResponse | Error message when Authy's response to the OTP token validation request is invalid. |
| OneTouchApprovalRequestCreationFailed | Error message when the OneTouch approval request failed. |
| InvalidOneTouchApprovalRequestCreationResponse | Error message when the response from OneTouch approval request creation is invalid. |
| OneTouchEndpointPollingFailed | Error message when polling for OneTouch status at the endpoint fails. |
| InvalidOneTouchEndpointResponse | Error message when the OneTouch status polling returns an invalid response. |
| InvalidOneTouchApprovalRequestStatus | Error message when the OneTouch status polling returns an invalid status. |
| NoOneTouchApprovalRequestStatus | Error message when the OneTouch status does not exist. |

### 9.3 Sample Message File

[Messages]
SplittingPassword = Separating password and OTP at delimiter '%s'
PlacingPasswordAndOTPIntoRequest = Placing password and OTP into request
CheckingStateCompatibility = Checking whether or not the state pertains to Authy authentication
UpdatingAuthTypeToReply = Valid Authy MFA state found; updating Auth-Type to '%s'

PerformingSilentAuthn = Performing silent authentication
PerformingInteractiveAuthnWithoutState = Performing interactive authentication with no pre-existing state
PerformingInteractiveAuthnWithState = Performing interactive authentication with the pre-existing state
AskingForAuthnMethod = Asking user for choice of authentication method

InvalidAuthnMethod = Unrecognized authentication method '%s'
MarkingFailedAuthnAttempt = Marking failed authentication attempt
UnexpectedAuthyResponse = Authy returned an unexpected response code %d with the following data: %s

VerifyingOTP = Verifying OTP
SendingOTPRequest = Sending OTP request
ParsingOTPResponse = Parsing the OTP response
AskingForOTP = Asking user for OTP
ParsingOTPVerificationResponse = Parsing OTP verification response
OTPAccepted = OTP accepted
OTPRejected = OTP rejected

CreatingOneTouchApprovalRequest = Creating OneTouch approval request
ParsingOneTouchApprovalRequestCreationResponse = Parsing OneTouch approval request creation response
PollingOneTouchEndpoint = Polling OneTouch approval request status endpoint
OneTouchApprovalRequestApproved = OneTouch approval request approved
OneTouchApprovalRequestDenied = OneTouch approval request denied
OneTouchApprovalRequestExpired = OneTouch approval request expired

ValidatingAuthzRequest = Validating authorization request
ValidatingAuthnRequest = Validating authentication request
ValidatingOTP = Validating OTP
ValidatingState = Validating the state data
ValidatingAuthnMethodChoice = Validating method choice

EnterAuthnMethod = <<EOT
Please choose an authentication method.
Enter 1 for OTP (token) or 2 for OneTouch.
EOT

ReenterAuthnMethod = <<EOT
Invalid authentication method specified.

Please choose an authentication method.
Enter 1 for OTP (token) or 2 for OneTouch.
EOT

EnterAuthnMethodAfterOTP = <<EOT
Incorrect OTP token.

Please choose an authentication method.
Enter 1 for OTP (token) or 2 for OneTouch.
EOT

```
EnterAuthnMethodAfterOneTouch = <<EOT
OneTouch approval request expired.

Please choose an authentication method.
Enter 1 for OTP (token) or 2 for OneTouch.
EOT

EnterOTP = Please enter your Authy token.
ReenterOTP = Please re-enter your Authy token.
AuthnSucceeded = Authentication succeeded
AuthnFailed = Authentication failed

[Errors]
InvalidConfigInt = Configuration option '%s/%s' has value '%s' which is not a valid
integer
InvalidConfigBool = Configuration option '%s/%s' has value '%s' which is not a valid
boolean
IDAndOTPParamsConflict = ID parameter and OTP parameter names must differ
NoProductionAPIKey = No production API key specified
NoSandboxAPIKey = No sandbox API key specified
InvalidProductionAPIKeyVerificationResponse = Could not process production API key
verification response: %s
InvalidSandboxAPIKeyVerificationResponse = Could not process sandbox API key
verification response: %s
ProductionAPIKeyVerificationFailed = Could not verify production API key: %s
SandboxAPIKeyVerificationFailed = Could not verify sandbox API key: %s
InvalidProductionAPIKey = Invalid production API key
InvalidSandboxAPIKey = Invalid sandbox API key
InvalidMaxAttemptCount = Max attempt count must be at least 1
NoAuthnMethods = No authentication methods enabled
NoOTPOption = No OTP authentication option value specified
NoOneTouchOption = No OneTouch authentication option value specified
OTPAndOneTouchOptionsConflict = OTP and OneTouch authentication option values
must differ
NoOTPDelimiter = No OTP delimiter specified
InvalidOneTouchPollingInterval = OneTouch approval request status polling interval
must be at least 1
InvalidOneTouchApprovalRequestTimeout = OneTouch approval request seconds-to-
live must be at least 0
NoOneTouchApprovalRequestMessage = No OneTouch approval request message
specified
NoOneTouchDefaultLogoURL = No default OneTouch approval request logo URL
specified
```

InvalidState = Invalid state data '%s'

UnexpectedOTPParam = Authy OTP parameter not expected
NoUserNameInRequest = No username specified in the request
NoPasswordInRequest = No password specified in the request
NoIDInRequest = No Authy ID specified in the request
NoOTPInRequest = No Authy OTP specified in the request
NoChallengeResponseInRequest = No challenge response in the request

InvalidOTP = Authy token must be a 7-digit code
OTPRequestFailed = Authy token request failed: %s
InvalidOTPResponse = Invalid response returned from OTP request: %s
OTPVerificationRequestFailed = Authy token verification request failed: %s
InvalidOTPVerificationResponse = Invalid response returned from Authy token verification request: %s

OneTouchApprovalRequestCreationFailed = OneTouch approval request creation failed: %s
InvalidOneTouchApprovalRequestCreationResponse = Invalid response returned from OneTouch approval request creation: %s
OneTouchEndpointPollingFailed = Polling OneTouch endpoint failed: %s
InvalidOneTouchEndpointResponse = Invalid response returned from OneTouch endpoint: %s
InvalidOneTouchApprovalRequestStatus = OneTouch endpoint returned unrecognized approval request status '%s'
NoOneTouchApprovalRequestStatus = No status returned from OneTouch endpoint

# 10 Additional Usage Notes

## 10.1 One configuration “style” per FreeRadius installation

A single FreeRadius server should only have one configuration of the modules mentioned in this document. For example, the using a single FreeRadius configuration for a client that supports challenge responses and another client that does not support challenge responses would not be recommended.