Installation Guide

FreeRadius Authy Multifactor Authentication

Table of Contents

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
David Curren		Enterprise Architect	1.0	12/22/2016

Revision History

Version	Issue Date	Description of Version/Changes	Author
0.1	12/07/2016	Initial draft	Toshie Takahashi
1.0	12/22/2016	Initial release	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. This document assumes the administrators have familiarity with the FreeRadius functionality and will be able to make decisions to the configuration if there are any conflicts with the in-place system. This document will also assume the appropriate administrators will be able to configure a VPN solution to utilize FreeRadius.

3 Overview

3.1 Authy MFA Module

The Authy Multi-factor Authentication (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 Authy ID Modules

A prerequisite to performing MFA with Authy is to retrieve the Authy ID to generate requests and validate tokens. The retrieval of the Authy ID is handled by the ID Store modules. Currently the LDAP and Flatfile modules are available. If an LDAP object stores a user's Authy ID the LDAP module should be used. If a comma separated values (CSV) file is used then the flatfile module should be used.

3.3 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 **Note**: The FreeRadius module should be compiled with the LDAP module enabled if LDAP is the target user store.
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

4.3 Perl Packages

The Authy MFA module requires the following perl packages installed via cpan.

- Config::IniFiles
- JSON
- HTML::HeadParser
- LWP::Protocol::https
- LWP::UserAgent
- IO::Socket::INET
- Net::LDAP
- Net::LDAPS
- ResourcePool
- ResourcePool::Factory::Net::LDAP
- Parse::CSV

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/authy**. If it does not exist, create the directory.

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

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

```
authy/
        config.ini
        Authy/
                AuthyAuthenticator.pl
                AuthyState.pm
                Configure.pm
                IDStores/
                        CSV.pm
                        LDAP.pm
                ModuleUtil.pm
                Text.pm
        messages.ini
```

6.2 Configuring FreeRadius

1. Ensure the FreeRadius server is shutdown.
2. Navigate to **mods-available**.

```
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/**.

```
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 {
        perl_flags = -I${modconfdir}/authy
        filename = ${modconfdir}/authy/Authy/AuthyAuthenticator.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-available/<SITE_NAME> if it does not exist already.

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

13. Edit the <SITE_NAME> file.

```
vim <SITE_NAME>
```

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

```
authorize {
        …
        update control {
                Auth-Type := "mfa"
        }
        authy
        …
}
```

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

```
authenticate {
        …
        Auth-Type mfa {
                …
                if (ok) {
                        authy
                }
        }

        Auth-Type authy-reply {
```

```
                authy
        }
        …
}
```

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

```
Auth-Type mfa{
        ldap
        if(ok) {
                authy
        }
}
```

16. Save and exit the file.
17. Navigate to **raddb**.

```
cd <FREERADIUS_HOME>/etc/raddb/
```

18. Edit the **dictionary** file.

```
vi dictionary
```

19. Insert the following lines. Ensure the numbers do not conflict with any other existing dictionary lines.

Note: This step is necessary to use custom request values. If the configuration for **IDParam** or **OTPParam** is changed from defaults, this file also must be changed. The below lines assume default configuration.

```
ATTRIBUTE Authy-ID      3500  string
ATTRIBUTE Authy-OTP     3501  string
```

20. Start the FreeRadius server. Ensure deployment is successful in the FreeRadius logs.

7 Callback Server (Recommended)

Authy provides two ways to validate a OneTouch request status.

- Polling directly to the Authy servers
- 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 the GET method.
8. Configure the **CustomPollingEndpoint** value in the Authy MFA module's configuration file to point to the callback host.
9.

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 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.

8.1.1 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. Ensure the value specified in this configuration exists in the FreeRadius **dictionary** file as an attribute. Configuration of this field can be left empty. **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. Ensure the value specified in this configuration exists in the FreeRadius **dictionary** file as an attribute.	No

	Configuration of this field can be left empty. **Default Value**: Authy-OTP	
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. Configuration of this field can be left empty. **Default Value**: authy-reply	No
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. Configuration of this field can be left empty. **Default Value**: Authy::AuthyState	No

8.1.2 Auth

Configuration Name	Description	Change Required
APIKeyEnv	The environmental variable name that will store the Authy API key. Configuration of this field can be left empty. **Default Value**: AUTHY_API_KEY	ENV
CompanyName	The name of the organization installing the FreeRadius module. This value will be used to send User-Agent headers when making requests to Authy. Configuration of this field is not required but strongly recommended to configure. **Example**: Example.com	ENV
Interactive	**True** if the client supports challenge responses. **False** if the client does not support challenge responses. Configuration of this field can be left empty.	ENV

	Default Value: False	
MaxAttempts	The maximum number of OTP attempts should be made before the client is responded with a REJECT response. Configuration of this field can be left empty. **Default Value**: 1	ENV
OTPEnabled	**True** if OTP validation is the desired flow for multifactor authentication. **False** if OTP validation is not desired. If both **OTP** and **OneTouch** are enabled and **Interactive** is **True**, an extra challenge response will be sent to the client prompting the user to select which authentication method is desired. If both **OTP** and **OneTouch** are enabled and **Interactive** is **False**, the module will assume **OneTouch** flow if the password does not contain the delimiter and token. If the password contains the **delimiter** value and OTP token, the **OTP** flow will be used. If OTP is disabled, then all configurations in the OTP section will be ignored. Configuration of this field can be left empty if OTP is not being used. **Default Value**: False	ENV
OneTouchEnabled	**True** if OneTouch validation is the desired flow for multifactor authentication. **False** if OneTouch validation is not desired. If both **OTP** and **OneTouch** are enabled and **Interactive** is **True**, an extra challenge response will be sent to the client prompting the user to select which authentication method is desired. If both **OTP** and **OneTouch** are enabled and **Interactive** is **False**, the module will assume **OneTouch** flow if the password does not contain the delimiter and token. If the password contains the **delimiter** value and OTP token, the **OTP** flow will be used. If OneTouch is disabled, then all configurations in the OneTouch section will be ignored.	ENV

	Configuration of this field can be left empty if OneTouch is not being used. **Default Value**: False	
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**. Configuration of this field is required if both OTP and OneTouch are enabled. **Example**: 1	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**. Configuration of this field is required if both OTP and OneTouch are enabled. **Example**: 2	ENV
IDStoreHome	The location on the filesystem the IDStore mapper module can be found. Configuration of this field is only required if the module is not stored in the default FreeRadius **mods-config/authy** directory. **Example**: /opt/custom_modules/	Yes
IDStoreModule	The module name used to import the IDStore mapper module. If using the out-of-the-box LDAP mapper module, specify value as **Authy::IDStores::LDAP**. If using the out-of-the-box flatfile mapper module, specify value as **Authy::IDStores::CSV**. Configuration of this field is required. **Example**: Authy::IDStores::LDAP	Yes

8.1.3 OTP

Configuration Name	Description	Change Required

Delimiter	This configuration is only used **Interactive** in the **Auth** section is set to **False**. 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. Configuration of this field can be left empty. **Default Value**: ,	No
Length	This configuration will be used to determine if the OTP provided is of the expected token length. Configuration of this field can be left empty. **Default Value**: 7	No
UseSandboxAPI	**True** if the Authy sandbox API endpoint will be used. An appropriate API key for the sandbox environment should also be set as the **AUTHY_API_KEY** environmental variable. **False** if the Authy production API endpoint will be used. An appropriate API key for the production environment should also be set as the **AUTHY_API_KEY** environmental variable. Configuration of this field can be left empty. **Default Value**: False	ENV
AlwaysSendSMS	**True** to send the OTP token via SMS. **False** to send only a push notification if the user's phone is a smartphone with Authy installed. Configuration of this field can be left empty. **Default Value**: False	ENV
AllowUnregisteredUsers	**True** to automatically accept users registered to Authy but not yet the application. **False** to deny access to unregistered users. Configuration of this field can be left empty. **Default Value**: False	ENV

8.1.4 OneTouch

Configuration Name	Description	Change Required
UseSandboxAPI	**True** if the Authy sandbox API endpoint will be used. An appropriate API key for the sandbox environment should also be set as the **AUTHY_API_KEY** environmental variable. **False** if the Authy production API endpoint will be used. An appropriate API key for the production environment should also be set as the **AUTHY_API_KEY** environmental variable. Configuration of this field can be left empty. **Default Value**: False	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. Configuration of a callback server is recommended if performance is a concern. Configuration of this field can be left empty if a callback server is not used.	Yes
VerifyCustomPollingEndpointHostname	**True** if hostname verification of custom callback server is desired. **False** if hostname verification of custom callback server is not desired. If **VerifyCustomPollingEndpointHostname** is disabled, then **CustomPollingEndpointCAFile** and **CustomPollingEndpointCAPath** values will be ignored. This field is only used if an HTTPS callback server is used. **Default Value**: True	ENV
CustomPollingEndpointCAFile	The location of the CA file containing the callback server certificate. Only required if the callback server certificate is not in the system wide certificate store. Otherwise leave this	ENV

	configuration empty. Use this configuration if the server certificate store is stored in a CA file. This configuration is only used if a **CustomPollingEndpoint** is configured and **VerifyCustomPollingEndpointHostname** is set to **True**. Otherwise this configuration can be left empty. **Example**: /opt/cafile	
CustomPollingEndpointCAPath	The directory location containing the callback server certificate. Only required if the callback server certificate is not in the system wide certificate store. Otherwise leave this configuration empty. Use this configuration if the server certificate is stored as a file in a directory. This configuration is only used if a **CustomPollingEndpoint** is configured and **VerifyCustomPollingEndpointHostname** is set to **True**. Otherwise this configuration can be left empty. **Example**: /opt/certs/	ENV
PollingInterval	The interval in seconds to wait between polling requests for OneTouch status. Configuration of this field can be left empty. **Default Value**: 0.5	ENV
ApprovalRequestTimeout	The expiration time in seconds to set for OneTouch approval requests. Configuration of this field can be left empty. **Default Value**: 86400	ENV
ApprovalRequestMessage	The message to send to the User along with the OneTouch request. Configuration of this field is required if **OneTouchEnabled** is **True**.	Yes
DefaultLogoURL	The URL to the default logo to display to users	ENV

	in the OneTouch request. If this configuration is not set, the image from the Authy dashboard will be utilized. If there is no image stored in the Authy dashboard, then no image will be displayed. 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. If this configuration is not set, no custom low-resolution image will be displayed to the user. Configuration of this field can be left empty.	ENV
MedResLogoURL	The URL to the normal resolution logo to display to users in the OneTouch request. If this configuration is not set, no custom normal resolution image will be displayed to the user. Configuration of this field can be left empty.	ENV
HighResLogoURL	The URL to the high-resolution logo to display to users in the OneTouch request. If this configuration is not set, no custom high-resolution image will be displayed to the user. Configuration of this field can be left empty.	ENV

8.1.5 IDStore

8.1.5.1 LDAP Mapper

Configuration Name	Description	Change Required
URI	The LDAP URI. This value should be of the form **ldap(s)://<host>:<port>**. **Note**: The use of either LDAPS or StartTLS-enhanced LDAP is strongly recommended.	Yes

	Configuration of this field is required. **Example**: ldaps://example.com:636	
UseStartTLS	**True** if connections to the LDAP server should use the StartTLS mechanism to elevate from insecure to secure. **False** if connections to the LDAP server should remain insecure/secure from start to finish. This field should only be set to **True** if the LDAP server supports StartTLS on the desired port. **Default Value**: False	
VerifyHostname	**True** if hostname verification of the LDAP is desired. **False** if hostname verification of LDAP is not desired. If **VerifyHostname** is disabled, then **CAFile** and **CAPath** values will be ignored. This field is only used for secure connections. **Default Value**: True	Yes
CAFile	The location of the CA file containing the LDAP server certificate. Only required if the callback server certificate is not in the system wide certificate store. Otherwise leave this configuration empty. Use this configuration if the server certificate store is stored in a CA file. Configuration of this field is required if **VerifyHostname** is set to **True**.	ENV
CAPath	The directory location containing the LDAP server certificate. Only required if the callback server certificate is not in the system wide certificate store. Otherwise leave this configuration empty. Use this configuration if the server certificate is stored as a file in a directory. Configuration of this field is required if **VerifyHostname** is set to **True**.	ENV
BindDN	The account to connect to LDAP with to retrieve	Yes

	the AuthyID attribute value. The use of a service account instead of an administrative account is strongly recommended. Configuration of this field is required. **Example**: cn=authysvc,dc=example,dc=com	
BindPasswordEnv	The environmental variable name that will store the **BindDN** password. Configuration of this field can be left empty. **Default Value**: LDAP_BIND_PASSWORD	No
UserBaseDN	The most specific DN containing all the users that should be able to utilize Authy multifactor authentication. Configuration of this field is required. **Example**: ou=Users,dc=example,dc=com	Yes
UserNameAttribute	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**. Configuration of this field is required. **Example**: uid	Yes
IDAttribute	The user attribute in LDAP storing the AuthyID attribute. Configuration of this field is required. **Example**: authyId	Yes
InitialConnectionPoolSize	The initial size of the LDAP server connection pool. This determines the number of connections to the LDAP server that will be opened when the FreeRADIUS server is first started. If the authentication load requires more concurrent connections than specified here, then the pool may open more connections. See **MaxConnectionPoolSize** for more information. **Default Value**: 2	No
MaxConnectionPoolSize	The maximum size of the LDAP server connection pool. This specifies the number of	No

	connections that may be opened and used on the LDAP server simultaneously. If this limit has been reached (i.e., this many connections are being used simultaneously by authentication requests), an authentication request that requires an LDAP connection will wait until another request is finished using its LDAP connection. **Default Value**: 5	
ConnectionRetryDelay	The delay in seconds between reconnection requests if a particular connection fails. This is generally set to allow a server to "wake up" in case a connection has failed due to a temporary outage. If this field is not set or set to zero, then a failed connection will try to reconnect immediately. **Default Value**: 0	No

8.1.5.2 Flatfile Mapper

Configuration Name	Description	Change Required
File	The name of the file AuthyID will be stored in. Configuration of this field is required.	Yes
UserNameColumnNumber	The column number that will store the username field. The AuthyID will be retrieved from the row containing the username. Configuration of this field is required. **Example**: 1	Yes
IDColumnNumber	The column number containing the AuthyID value for the user. Configuration of this field is required. **Example**: 2	Yes
Separator	The value indicating a separator of columns in the file. In a comma-separated value (CSV) file, this is generally a comma.	Yes

	Configuration of this field can be left empty. **Default Value**: ,	
Quote	The character indicating a quote character in the file. Configuration of this field can be left empty. **Default Value**: “	Yes
EscapeCharacter	The character indicating the quote escape character, i.e., the character used to escape quotes in the file. The default value is “, meaning that “” within a quoted string will be interpreted as “. For example, the quoted string “Hello, “”John””.” Is interpreted as: Hello, “John”. Configuration of this field can be left empty. **Default Value**: “	Yes

8.2 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
APIKeyEnv	Any string value	
CompanyName	Any string value	
Interactive	True/False	
MaxAttempts	Any positive integer value	
OTPEnabled	True/False	
OneTouchEnabled	True/False	
OTPOption	Any string value	
OneTouchOption	Any string value	
IDStoreHome	Any string value	
IDStoreModule	Any string value	

OTP		
Configuration Name	Accepted Values	Environment Value
Delimiter	Any string value	
Length	Any positive integer value	
UseSandboxAPI	True/False	
AlwaysSendSMS	True/False	
AllowUnregisteredUsers	True/False	

OneTouch		
Configuration Name	Accepted Values	Environment Value
UseSandboxAPI	True/False	
CustomPollingEndpoint	Any URL string value	
VerifyCustomPollingEndpoi	True/False	

ntHostname		
CustomPollingEndpointCAFile	Any string value	
CustomPollingEndpointCAPath	Any string value	
PollingInterval	Any positive decimal value	
ApprovalRequestTimeout	Any positive 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	
UseStartTLS	True/False	
VerifyHostname	True/False	
CAFile	Any string value	
CAPath	Any string value	
BindDN	Any string value	
BindPasswordEnv	Any string value	
UserBaseDN	Any string value	
UserNameAttribute	Any string value	
IDAttribute	Any string value	
InitialConnectionPoolSize	Any positive integer value	
MaxConnectionPoolSize	Any positive integer value	
ConnectionRetryDelay	Any non-negative integer value	

IDStore (Flatfile Mapper)		
Configuration Name	Accepted Values	Environment Value
File	Any string value	
UserNameColumnNumber	Any positive integer value	
IDColumnNumber	Any positive integer	

	value	
Separator	Any string value up to 8 bytes	
Quote	Any string value up to 8 bytes	
EscapeCharacter	Any single character	

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]

[Auth]
CompanyName = example.com
Interactive = True
MaxAttempts = 3
OTPEnabled = True
IDStoreModule = Authy::IDStores::LDAP

[OTP]

[OneTouch]

[IDStore]
URI = ldaps://example.com:636
VerifyHostname = True
CAFile = /tmp/cafile
BindDN = cn=Directory Manager
UserBaseDN = dc=example,dc=com
UserNameAttribute = 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]

[Auth]
CompanyName = example.com
Interactive = False
OTPEnabled = True
IDStoreModule = Authy::IDStores::LDAP

[OTP]

[OneTouch]
```

```
[IDStore]
URI = ldaps://example.com:636
VerifyHostname = True
CAFile = /tmp/cafile
BindDN = cn=Directory Manager
UserBaseDN = dc=example,dc=com
UserNameAttribute = 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]

[Auth]
CompanyName = example.com
Interactive = False
OneTouchEnabled = True
IDStoreModule = Authy::IDStores::LDAP

[OTP]

[OneTouch]
CustomPollingEndpoint = https://callback.example.com/polling
VerifyCustomPollingEndpointHostname = True
CustomPollingEndpointCA = /tmp/cafile
PollingInterval = 0.5
ApprovalRequestTimeout = 86400
ApprovalRequestMessage = FreeRadius has requested your approval.

[IDStore]
URI = ldaps://example.com:636
VerifyHostname = True
CAFile = /tmp/cafile
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]
```

```
[Auth]
CompanyName = example.com
Interactive = True
MaxAttempts = 5
OTPEnabled = True
OneTouchEnabled = True
OTPOption = 1
OneTouchOption = 2
IDStoreModule = Authy::IDStores::LDAP

[OTP]
Enabled = True

[OneTouch]
Enabled = True
UseSandboxAPI = False
CustomPollingEndpoint = https://callback.example.com/polling
PollingInterval = 0.5
ApprovalRequestTimeout = 86400
ApprovalRequestMessage = FreeRadius has requested your approval.

[IDStore]
URI = ldaps://example.com:636
VerifyHostname = True
CAFile = /tmp/cafile
BindDN = cn=Directory Manager
UserBaseDN = dc=example,dc=com
UserNameAttribute = uid
IDAttribute = authyId
```

8.3.5 OneTouch and TOTP Authentication in Silent mode

```
[RADIUS]

[Auth]
CompanyName = example.com
Interactive = False
MaxAttempts = 5
OTPEnabled = True
OneTouchEnabled = True
OTPOption = 1
OneTouchOption = 2
IDStoreModule = Authy::IDStores::LDAP

[OTP]
Enabled = True
```

```
[OneTouch]
Enabled = True
UseSandboxAPI = False
CustomPollingEndpoint = https://callback.example.com/polling
PollingInterval = 0.5
ApprovalRequestTimeout = 86400
ApprovalRequestMessage = FreeRadius has requested your approval.

[IDStore]
URI = ldaps://example.com:636
VerifyHostname = True
CAFile = /tmp/cafile
BindDN = cn=Directory Manager
UserBaseDN = dc=example,dc=com
UserNameAttribute = 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 unless silent mode is used.
3. 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 User Prompts

Message Name	Description
EnterMethod	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. If the user is re-prompted for an authentication method, this message is appended to the end of **InvalidMethod**, **IncorrectOTP**, or **OneTouchExpired**, depending on the reason. A blank line will

	separate the two messages.
InvalidMethod	Text snippet prepended to **EnterMethod** (separated by a blank line) and sent to client if the user has provided an invalid authentication method. This message is only sent if the user has at least one authentication attempt remaining. Ensure that the user provides one of the values set in the **OTPOption** or **OneTouchOption** configuration.
EnterOTP	Message sent to client to request an OTP from the user during a challenge-response OTP authentication flow. If the user is re-prompted for an OTP (meaning only OTP authentication is enabled), this message is appended to **IncorrectOTP**. A blank line will separate the two messages.
IncorrectOTP	Text snippet prepended to **EnterMethod** or **EnterOTP** (separated by a blank line) and sent to the client if the user has provided an invalid OTP during a challenge-response OTP authentication flow. This message is only sent if the user has at least one authentication attempt remaining and OneTouch authentication is also enabled.
OneTouch	Messages sent to user's phone in a OneTouch approval request. **Note:** This message will be seen on the Authy app itself, not the client.
OneTouchExpired	Text snippet prepended to **EnterMethod** and sent to the client if the user's OneTouch approval request has expired during a challenge-response OneTouch authentication flow. This message is only sent if the user has at least one authentication attempt remaining and OTP authentication is also enabled.

9.1.2 Authentication Results

Message Name	Description
Succeeded	Message sent to client when the authentication is deemed a success.
Failed	Message sent to client when the

	authentication is deemed a failure.
NoID	Message sent to the client when the user's Authy ID cannot be determined, implying that the user may not have Authy authentication configured for their account.
Error	Message sent to client when the authentication fails for a reason outside of the user's control. Error details can be accessed in the RADIUS server logs, typically by a system administrator.

9.2 Sample Message File

[Prompts]
EnterMethod = <<EOT
Please choose an authentication method.
Enter 1 for OTP (token) or 2 for OneTouch.
EOT
InvalidMethod = Invalid authentication method specified.
EnterOTP = Please enter your Authy token.
IncorrectOTP = Incorrect Authy token.
OneTouch = Login requested.
OneTouchExpired = Authy OneTouch approval request expired.

[Results]
Succeeded = Authentication succeeded.
Failed = Authentication failed.
NoID = Your account does not appear to be configured for Authy authentication. Please contact your System Administrator to resolve the issue.
Error = A system error occurred during authentication. Please contact your System Administrator to resolve the issue.

10 Troubleshooting

10.1 ID Store Errors

Error Code	Description
00-001	**Module load failed** The ID store module could not be loaded for some reason. The Perl-provided error message is included.
00-002	**Invalid module**

	The ID store module does not implement the required subroutines.
00-003	**Initialization failed** The ID store initialization process failed for some reason. The error message provided by the ID store is included.

10.2 Request Errors

Error Code	Description
01-001	**No username in request** Username is not found in the request. Ensure the user has provided a username in the initial prompt by the client. Ensure other FreeRADIUS modules have not removed the username value from the request.
01-002	**No password in request** Password is not found in the request. Ensure the user has provided a password in the initial prompt by the client. Ensure other FreeRADIUS modules have not removed the password value from the request.
01-003	**No Authy ID in request** Authy ID was not found in the request. Ensure the ID Store module can correctly retrieve Authy ID from the target user store.
01-004	**No OTP in request** OTP was not found in the request. Ensure the user has provided an OTP to the challenge response or in the initial response, depending on the configuration. Ensure other FreeRadius modules have not removed the OTP value from the request.
01-005	**No challenge response in request** The content from a challenge response is empty. Ensure the user has provided a response to any challenges. Ensure other FreeRadius modules have not removed a challenge response from the request.
01-006	**Unexpected OTP Parameter** The OTP request parameter was found in the FreeRADIUS request prematurely, i.e., before the authenticator processed the request itself.

01-007	**ID retrieval failed** The ID store retrieval module could not retrieve an Authy ID. Ensure the target ID store contains the user's Authy ID. Ensure the ID store configuration is correct for the module.
01-008	**Invalid ID** The retrieved Authy ID is invalid. Ensure the stored ID is valid or the retrieved attribute from the ID store is correct.
01-009	**No ID found for user** Authy ID could not be found for the user in the user store. Ensure the Authy ID is stored in the target ID store.

10.3 OTP Errors

Error Code	Description
02-001	**Prompt Request Failed Internally** Sending the request to Authy to provide the user with an OTP has failed before reaching the Authy server. Ensure the FreeRadius server is able to communicate with the Authy server.
02-002	**Prompt Request Failed Externally** Sending the request to Authy to provide the user with an OTP has failed after reaching the Authy server. The Authy-provided error message is included.
02-003	**Verification Request Failed Internally** Sending the request to Authy to verify the OTP has failed before reaching the Authy server. Ensure the FreeRadius server is able to communicate with the Authy server.
02-004	**Verification Request Failed Externally** Sending the request to Authy to verify the OTP has failed after reaching the Authy server. The Authy-provided error message is included.

10.4 OneTouch Errors

Error Code	Description
03-001	**Approval Request Creation Failed Internally** Sending the request to Authy to create a OneTouch approval has failed before reaching the Authy server. Ensure the FreeRadius server is able to communicate with the Authy server.
03-002	**Approval Request Creation Failed Externally** Sending the request to Authy to create a OneTouch approval has failed after reaching the Authy server. The Authy-provided error message is included.
03-003	**API Endpoint Failed Internally** Sending the status request to the Authy polling endpoint has failed before reaching the Authy server. Ensure the FreeRadius server is able to communicate with the Authy server.
03-004	**API Endpoint Failed Externally** Sending the status request to the Authy polling endpoint has failed after reaching the Authy server. The Authy-provided error message is included.
03-005	**Custom Endpoint Failed Internally** Sending the status request to the custom polling endpoint has failed before reaching the custom server. Ensure the FreeRadius server is able to communicate with the custom server.
03-006	**Custom Endpoint Failed Externally** Sending the status request to the custom polling endpoint has failed after reaching the custom server. The response data from the custom endpoint is included.
03-007	**Endpoint Returned Invalid Status** The approval request status from the endpoint is not an accepted value.

11 Additional Usage Notes

11.1 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.

12 Custom ID Store Modules

The code provided presents a simple way to integrate a custom Authy ID retrieval module with the existing code. The current modules available enable retrieval of an Authy ID from an LDAP or Flatfile. The requirements for a custom retrieval module are as follows:

- An **initialize** subroutine is defined.
- A **get_authy_id** subroutine is defined that returns the value of a given user's Authy ID. The username provided in login is always provided to the subroutine. If no Authy ID could be retrieved, expect a null return.
- (Optional) **destroy** subroutine is defined.

Once the module is created, edit the configuration file's **IDStoreHome** and **IDStoreModule** values as necessary to point to the custom module.

13 Custom Callback Server

The code provided as the callback server simply parses the callback request from Authy and extracts the OneTouch status. If a request comes in to the server with a particular UUID and associated API Key, then the server will provide the stored OneTouch status. As long as the following behaviors are met, a custom callback can also be implemented:

- Capable of handling HTTP GET or POST callbacks from Authy.
- Requests to the callback server are of form <host>/polling/<uuid> and has the following contents:
    - X-Authy-API-Key header containing the Authy API Key
    - UserAgent header containing **CompanyName** and system information
- Response to polling requests must be as follows:

HTTP Status	Content Type	Body	Description
200	text/plain	approved	Callback status indicated approval for UUID
200	text/plain	denied	Callback status indicated denial for UUID
204			The callback server has to received a callback from Authy for the requested UUID.

- Note the callback server does not handle expiration as Authy does not perform a callback for expired OneTouch requests.

14 External References

Document	URL
FreeRadius Wiki	http://wiki.freeradius.org/Home
Apache Tomcat 8.0 Documentation	http://tomcat.apache.org/tomcat-8.0-doc/index.html
OpenVPN RADIUS Plugin	http://www.nongnu.org/radiusplugin/
Cisco ASA 5512-X RADIUS Configuration	http://www.cisco.com/c/en/us/support/docs/security/asa-5500-x-series-next-generation-firewalls/98594-configure-radius-authentication.html **Note**: When configuring for Radius authentication and using OneTouch flows, ensure the VPN client timeout is set to a reasonable period of time to approve or reject a OneTouch request.