# 4TRESS™ AAA Server for Remote Access

## Web Authentication API Reference Manual

## Table of Contents


ALPHABETICAL LIST OF ASP FUNCTIONS . . . . . 3

ALPHABETICAL LIST OF ASP.NET FUNCTIONS . . . . . 5

CHAPTER 1: INTRODUCTION . . . . . 6

About the 4TRESS AAA Server for Remote Access . . . . . 6

About the Web Access Agent . . . . . 6

About the Web Authentication API (WAA) . . . . . 7

About this document . . . . . 7

CHAPTER 2: OVERVIEW . . . . . 8

Integration of WAA applications . . . . . 8

Instantiating the WAA component . . . . . 8

Sample code . . . . . 8

CHAPTER 3: ASP FUNCTIONS . . . . . 11

Summary . . . . . 11

General purpose services . . . . . 13

Session management services . . . . . 13

Authentication services . . . . . 14

RADIUS attributes management services . . . . . 18

CHAPTER 4: ASP.NET FUNCTIONS . . . . . 22

Summary . . . . . 22

General purpose services . . . . . 23

Session management services . . . . . 23

Authentication services . . . . . 24

RADIUS attributes management services . . . . . 27

# 4TRESS AAA Server for Remote Access

## Web Authentication API Reference Manual

### Alphabetical List of ASP Functions


Authenticate . . . . . 15

CheckUserID . . . . . 17

GetAttributeInt . . . . . 18

GetAttributeString . . . . . 18

GetCipheredAttribute . . . . . 19

GetUserName . . . . . 17

Initialize . . . . . 13

IsAuthenticated . . . . . 16

KillSession . . . . . 14

OnEndSession . . . . . 13

SetAttributeInt . . . . . 21

SetAttributeString . . . . . 20

SetSessionChallenge . . . . . 14

Trace . . . . . 13

# 4TRESS AAA Server for Remote Access

## Web Authentication API Reference Manual

### Alphabetical List of ASP.NET Functions


AuthenticateX . . . . . 25

GetAttributeIntX . . . . . 27

GetAttributeStringX . . . . . 27

GetCipheredAttributeX . . . . . 28

GetUserNameX . . . . . 26

Initialize . . . . . 23

IsAuthenticatedX . . . . . 26

KillSessionX . . . . . 23

SetSessionChallengeX . . . . . 24

Trace . . . . . 23

# Chapter 1: Introduction

## About the 4TRESS AAA Server for Remote Access

The 4TRESS AAA Server for Remote Access is a strong RADIUS, TACACS+ and IEEE 802.1x authentication server that maps to an existing LDAP directory to provide strong user authentication services for a wide range of access points.

AAA stands for:

- Authentication - accepts or rejects user authentication requests based on stored credentials and/or one-time passwords.
- Authorization - controls user access based on the appropriate attributes transmitted to the network remote access point (VPN, firewall, router etc.,).
- Accounting - stores information concerning user activity while connected remotely (connection times, data transfers etc.).

The AAA Server makes it possible to manage users through an LDAP directory without requiring LDAP schema extensions. Users authenticate through the AAA Server with smart cards, hardware and software tokens, USB keys, mobile devices, PDAs, (and optionally, with static or static LDAP passwords).

A secure remote access solution, the AAA Server enables you to protect network access methods such as:

- Web access
- Remote access via dial-up
- Remote access via VPN
- Remote desktop environments (Windows and Citrix)
- SSL VPN
- Wireless LAN access

## About the Web Access Agent

The Web Access Agent is a Radius/Tacacs plug-in for an IIS server. The Web Access Agent allows the server to communicate with an 4TRESS AAA Server for Remote Access, thus allowing use of SKI authentication on websites. The Web Access Agent supports both challenge response and synchronous authentication.

## Components in the Web Access Agent

The Web Access Agent consists of:

- WebHelpdesk
  Provides a console help desk through a Web interface.
- Web Access Agent
  This Radius/Tacacs plug-in allows an IIS server to communicate with an AAA Server, and thus supports both challenge/response and synchronous SKI authentication on websites.

  The Web Authentication API (WAA) is an ActiveX control that provides Web Access Agent functionality from Microsoft Active Server Pages (ASP and ASP.NET) Web scripts.

# About the Web Authentication API (WAA)

The Web Authentication API (WAA) provides a set of services that make it possible to secure access to websites or parts of a website using calls from Web applications.

The WAA API is distributed with the Web Access Agent within the ActivIdentity AAA Server authentication solution. It is designed on Microsoft Active Server Pages (ASP) and ActiveX Server Pages (ASPX) deployed in a Microsoft Internet Information Server environment.

Refer to the *Web Access Agent Installation and Configuration Guides* for information on how to install and configure the Web Access Agent.

## Prerequisites

WAAWeb Authentication APIis an ASP/ActiveX API. It works with ActivIdentity AAA Servers version 6.5.

# About this document

Readers of this document are assumed to be experienced programmers who:

- Know ASP/ActiveX.

The chapters in this document are:

- Chapter 1, "Introduction" on page 6
- Chapter 2, "Overview" on page 8
- Chapter 3, "ASP Functions" on page 11
- Chapter 4, "ASP.NET Functions" on page 22

# Chapter 2: Overview

## Integration of WAA applications

The integration of the API services is the responsibility of the Application Service Provider programmer. The following rules must be followed:

- The component is designed to be used in server-side scripts to provide strong authentication using ActivIdentity AAA Server. A number of interfaces are available to the Active Server Pages programmer.
- The `ProgID` of the component is '`ActivPack.WebAccessAgent`'.
- The component should have application scope and be instantiated in `global.asa` for ASP or `global.asax` for ASP.NET.
- The component supports both apartment-threaded and free-threaded threading models (and includes a free-threaded marshaller).
- The scripting language used in this section is JavaScript.

## Instantiating the WAA component

It is assumed in the code examples that the component has been instantiated in the following `global.asa` or `global.asax` function:

```
function Application_OnStart() {
  ActivPackWAA = Server.CreateObject("ActivPack.WebAccessAgent");
  Application("ActivPackWAA") = ActivPackWAA;
  ActivPackWAA.Initialize();
}
```

There are two additional services included for compatibility with Active Server Pages Components:

- `OnStartPage`
- `OnEndPage`

They do not implement any functionality.

## Sample code

### Samples for the ASP environment

The samples provide an example of how to use the WAA API with an ActiveX Server Component. Both samples demonstrate the use of a RADIUS attribute to redirect a user to their home page and to retrieve the authenticated username. Both methods can be configured in `global.asa`.

The samples demonstrate two different methods for authenticating users:

- Synchronous authentication
- Asynchronous challenge/response authentication

In the ActivIdentity AAA Server, the '`Web-Default-Page`' attribute is set to the LDAP parameter '`WWWHomePage`' of the authorization profile.

This parameter is set to the personal home page on the protected website of each user in the group.

When a user is successfully authenticated, the client browser redirects to the home page as defined by the link to the '`Web-Default-Page`' in the HTML form.

If the ActivIdentity AAA Server is not configured with an authorization profile that provides the home page name, the samples display the message:

```
"No 'Web-Default-Page' has been submitted by the Authentication server."
```

The samples provided are:

| Filename | Description |
|---|---|
| `global.asa` | Initialization parameters |
| `authForm.asp` | Synchronous authentication form |
| `authFormAsync.asp` | Asynchronous authentication form |
| `checkAuth.asp` | Check if the session is still authenticated |
| `killSession.asp` | Close the session |
| `logon.asp` | Start an authentication process |
| `mainForm.asp` | Protected page shows after authentication |

## Samples for the ASP.NET Environment

Some samples are provided to show an example use of the ActiveX Server Component. The samples use two different methods to authenticate a user:

- Synchronous authentication
- Asynchronous challenge/response authentication

Both methods can be configured in `global.asax`.

The samples demonstrate the use of a RADIUS attribute to redirect a user to their home page and to retrieve the authenticated username.

In the ActivCard AAA Server, the '`Web-Default-Page`' attribute is set to the LDAP parameter '`WWWHomePage`' of the authorization profile.

For each user in the group, this parameter is set to their personal home page on the protected web site.

When a user is successfully authenticated, the client browser redirects to the home page as defined by the link to the '`Web-Default-Page`' in the HTML form.

If the ActivCard AAA Server is not configured with an authorization profile that provides the home page name, the samples display the following message:

```
"No 'Web-Default-Page' has been submitted by the Authentication server."
```

The samples provided are:

| Filename | Description |
|---|---|
| `global.asax` | Initialization parameters |
| `authForm.aspx` | Synchronous authentication form |
| `authFormAsync.aspx` | Asynchronous authentication form |
| `checkAuth.aspx` | Check if the session is still authenticated |
| `killSession.aspx` | Close the session |
| `logon.aspx` | Start an authentication process |
| `mainForm.aspx` | Protected page shows after authentication |

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/395024114133011221