



# Retina Network Security Scanner

Scan Management Service Guide

Revision/Update Information: November 2016

Software Version: 6.0

Revision Number: 0

**Corporate Headquarters**

5090 N. 40<sup>th</sup> Street  
Phoenix, AZ 85018  
Phone: 1 818-575-4000

**COPYRIGHT NOTICE**

Copyright © 2016 BeyondTrust Software, Inc. All rights reserved.

The information contained in this document is subject to change without notice.

No part of this document may be photocopied, reproduced or copied or translated in any manner to another language without the prior written consent of BeyondTrust Software.

BeyondTrust Software is not liable for errors contained herein or for any direct, indirect, special, incidental or consequential damages, including lost profit or lost data, whether based on warranty, contract, tort, or any other legal theory in connection with the furnishing, performance, or use of this material.

All brand names and product names used in this document are trademarks, registered trademarks, or trade names of their respective holders. BeyondTrust Software is not associated with any other vendors or products mentioned in this document.

# Contents

- Introduction.....4**
- Contacting Support .....4
- Telephone ..... 4
- Privileged Account Management Support..... 4
- Vulnerability Management Support ..... 4
- All other Regions:..... 4
- Online ..... 4
- Using WCF Retina Scan Management Service.....5**
- Overview .....5
- Security.....5
- Configuration.....5
- Class Reference .....7
- Code Example.....8

## Introduction

Retina Network Security Scanner provides vulnerability testing for multiple platforms, automatic fixes of vulnerabilities and the ability to create your own audits.

This guide is intended for network security administrators who are using the Retina API and want to manage Retina scans remotely.

## Contacting Support

---

For support, go to our [Customer Portal](#) then follow the link to the product you need assistance with.

The Customer Portal contains information regarding contacting Technical Support by telephone and chat, along with product downloads, product installers, license management, latest product releases, product documentation, webcasts and product demos.

### Telephone

#### Privileged Account Management Support

Within Continental United States: 800.234.9072  
Outside Continental United States: 818.575.4040

#### Vulnerability Management Support

North/South America: 866.529.2201 | 949.333.1997  
+ enter access code

#### All other Regions:

Standard Support: 949.333.1995  
+ enter access code  
Platinum Support: 949.333.1996  
+ enter access code

### Online

<http://www.beyondtrust.com/Resources/Support/>

## Using WCF Retina Scan Management Service

The WCF Retina Scan Management service enables you to remotely manage Retina scans from any application that can send and consume a SOAP message.

This document contains the following sections:

- Overview
- Security
- Configuration
- Class Reference
- Code Example

### Overview

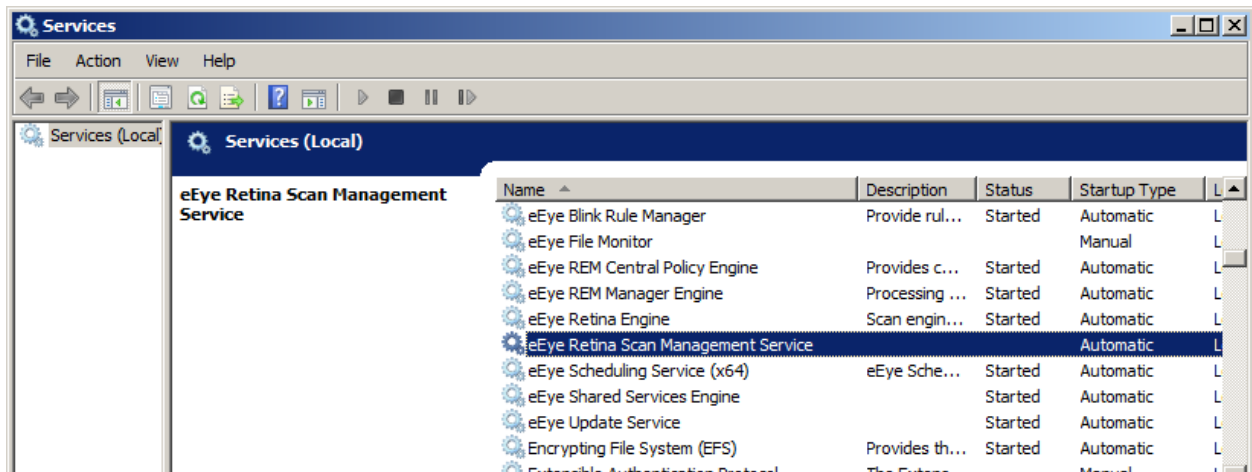
The WCF Retina Scan Management service provides a subset of features that are available in Retina Network Security Scanner. Use the service to manage Retina vulnerability and discovery scans from an application that is running remotely.

### Security

Retina Scan Management Service supports Windows based and Certificate based authentication.

### Configuration

Retina Scan Management Service is a self-hosting WCF Windows service. The service is installed with Retina:



By default the service is not started. Use the configuration tool to configure the service for your environment. The configuration tool is located in the following directory:

C:\Program Files\BeyondTrust\Retina 5\Retina.Service.Config.exe

```

Administrator: Command Prompt
Usage Retina.Service.Config.exe [enableservice | disableservice | setpermissions | exportcert] [switches]

Options:
  enableservice           Enable the Retina management service
    /authtype <windows|certificate>  Select the client authentication type
                                     Default: Windows.
    /port Port                Port to listen on. Default: 8980
    /certificatename CertificateName  Name of certificate to use for SSL
                                     Server bindings
                                     Default: "Retina Scan Manager Server"

  disableservice         Disable the Retina management service

  setpermissions         Set Windows users and groups that are
                         allowed to connect. These permissions
                         are only used if the authentication type
                         is set to Windows.
    /allowgroups group1,group2,...  List of Windows user groups allowed
                                     to connect. Separated by commas (,)
    /allowusers user1,user2,...     List of Windows users allowed to
                                     connect. Separated by commas (,)

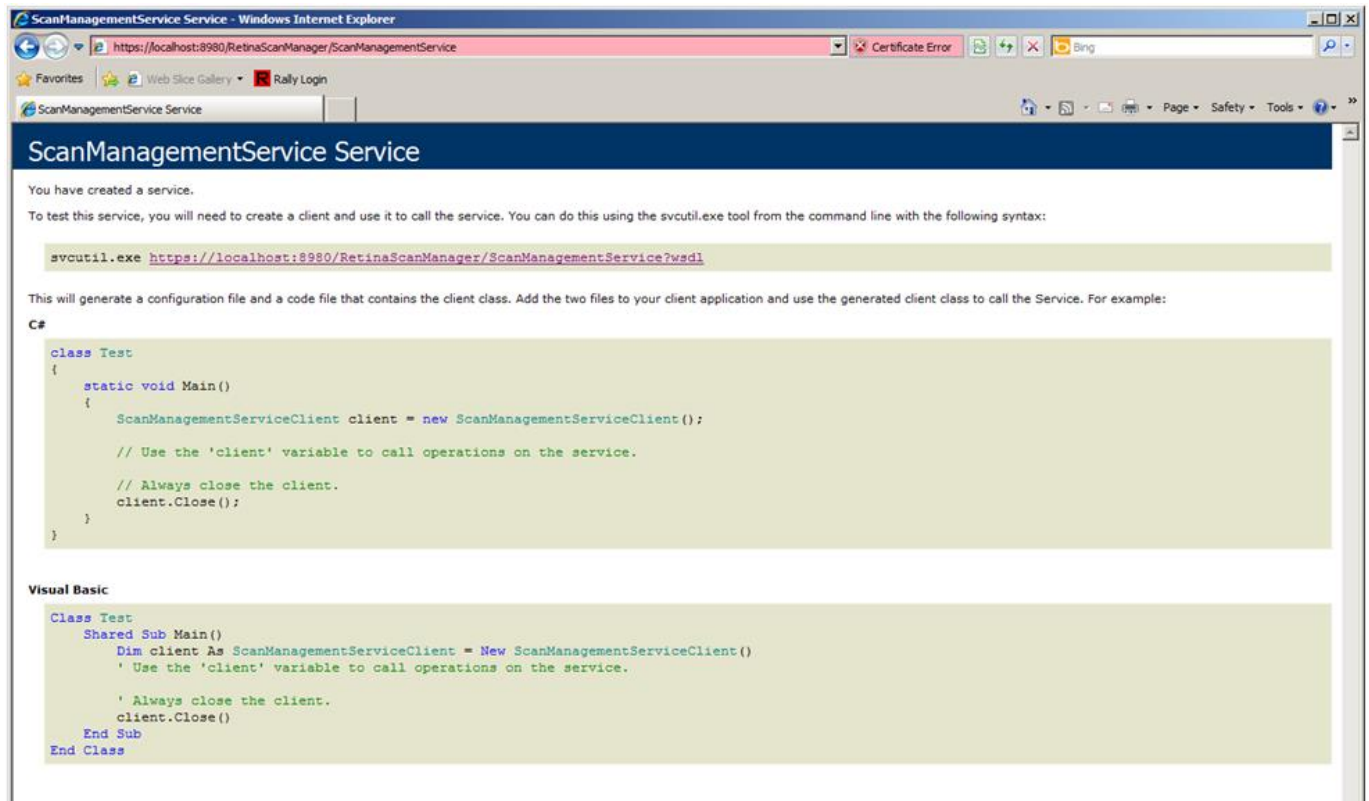
  exportcert FilePath Password  Export the certificate to a file for
                                   the client to use. This certificate is
                                   only needed if the authentication type
                                   is set to Certificate.

Example: Retina.Service.Config.exe enableservice

```

Once the service is enabled, you can verify the installation by going to:

<https://localhost:8980/RetinaScanManager/ScanManagementService>



```

ScanManagementService Service - Windows Internet Explorer
https://localhost:8980/RetinaScanManager/ScanManagementService
Certificate Error
ScanManagementService Service
ScanManagementService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

svcutil.exe https://localhost:8980/RetinaScanManager/ScanManagementService?wsdl

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

C#
class Test
{
    static void Main()
    {
        ScanManagementServiceClient client = new ScanManagementServiceClient();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}

Visual Basic
Class Test
    Shared Sub Main()
        Dim client As ScanManagementServiceClient = New ScanManagementServiceClient()
        ' Use the 'client' variable to call operations on the service.

        ' Always close the client.
        client.Close()
    End Sub
End Class

```

The wsdl file is also available here:

<https://localhost:8980/RetinaScanManager/ScanManagementService?wsdl>

## Class Reference

**Namespace:** Retina.Service.ScanManager

**Methods:**

Name	Description
Clean(UInt32 flags)	Clean Retina Queue, Schedule, Logs, Jobs
DeleteCredential(string description)	Remove credential
GetAuditInformation(int rthId)	Returns Retina Audit information based on retina RTHID
GetEngineInformation()	Returns Retina engine information
GetJobStatus(string scanID);	Returns Retina job status based on Scan ID
GetJobStatusAll()	Lists Retina jobs
GetPackageSupportForAllJobs()	Runs Package Support Tool to collect log files and downloads result package
GetPackageSupportForJob(string jobId, string groupId)	Runs Package Support Tool to collect log files for a specific scan job and downloads result package
GetProductInformation()	Returns Retina product information
GetScheduledJobs()	Lists scheduled Retina jobs
PauseScan(string scanID)	Pause a running scan based on Scan ID
ResumeScan(string scanID)	Resume a paused scan based on ScanID
SetCredential(string username, string password, string description)	Set credential to be used for Retina Scan
StartScan(string scanID)	Starts a scan based on ScanID
StopScan(string scanID)	Stop a scan based on ScanID
StopSchedule(string scanID)	Remove scheduled scan
UpdateConfiguration(ScanFileInfo fileInfo)	Updates Retina configuration information
UpdateRetina()	Downloads latest Retina and updates Retina
UpdateLicense(string serial, RegistrationInfo registrationInfo)	Updates Retina serial and license Registration information
UploadAddress(ScanFileInfo fileInfo)	Uploads an Address group file to be used for Retina Scan
UploadPort(ScanFileInfo fileInfo)	Uploads a Port group file to be used for Retina Scan
UploadAudits(ScanFileInfo fileInfo)	Uploads an Audit group file to be used for Retina Scan
UploadScanRequest(ScanFileInfo fileInfo)	Uploads a Scan Request file to be used for Retina Scan
UploadJob(ScanFileInfo fileInfo)	Uploads a Job file to be used for Retina Scan

**Fault:**

Name	Description
ValidationFault	When uploaded file (address, port, audit, scanrequest, job) is not valid, ValidationFault FaultException is thrown.

## Code Example

---

### Client Authentication:

```
// windows authentication
var client = new ScanManagementServiceClient();
client.ClientCredentials.Windows.ClientCredential.UserName = txtUserName.Text;
client.ClientCredentials.Windows.ClientCredential.Password = txtPassword.Text;

// certificate authentication
var client = new ScanManagementServiceClient();
client.ClientCredentials.ClientCertificate.SetCertificate(StoreLocation.LocalMachine,
StoreName.My,
X509FindType.FindBySubjectName,
"Retina Scan Manager Server");
```

### Scan Actions (Start, Stop, Pause, Resume, StopSchedule):

```
var client = new ScanManagementServiceClient();
result = client.StartScan(textBox1.Text);
```

### Set Scan Credential:

```
var client = new ScanManagementServiceClient();
client.SetCredential("administrator", "password", "scan credential");
```

### Upload Scan Files (Audit, Address, Port, ScanRequest, Job):

```
var fileInfo = new FileInfo("c:\\address.xml");
var client = new ScanManagementServiceClient();
using (var stream = new FileStream(FileTextBox.Text, FileMode.Open, FileAccess.Read))
{
    client.UploadAddress(fileInfo.Name, fileInfo.Length, true, stream);
}
```

Note: The schema file for scan files is stored under: C:\Program Files\BeyondTrust\Retina 5\Database\Schemas\  
If the uploaded file is not valid based on the schema file definition, a ValidationFault FaultException is thrown by the server.

### Clean Engine:

```
[Flags]
public enum eCleanFlags
{
    None = 0,
    QUEUE = 1,
    SCHEDULE = 2,
    LOGS = 4,
    JOBS = 8
}

eCleanFlags cleanFlags = eCleanFlags.None;
cleanFlags |= eCleanFlags.QUEUE;
cleanFlags |= eCleanFlags.SCHEDULE;
cleanFlags |= eCleanFlags.LOGS;
cleanFlags |= eCleanFlags.JOBS;

var client = new ScanManagementServiceClient();
client.Clean((uint) cleanFlags);
```

### List Job Status:

```
ScanManagementServiceClient client = getClient();
Dictionary<string, int> jobStatus = client.GetJobStatusAll();

foreach (var job in jobStatus)
{
    // loop through jobs
}
```



### Get Audit Information:

```
ScanManagementServiceClient client = getClient();
XElement auditData = client.GetAuditInformation(Int32.Parse(txtRthID.Text));
```

### Get Product Information:

```
ScanManagementServiceClient client = getClient();
ProductInfo productInfo = client.GetProductInformation();

string productInfotext = "Audit Version:" + productInfo.AuditVersion;
productInfotext += Environment.NewLine;

productInfotext += "Engine Version: " + productInfo.EngineVersion;
productInfotext += Environment.NewLine;

productInfotext += "IP Count:" + productInfo.IPCount;
productInfotext += Environment.NewLine;

productInfotext += "IP Count Text:" + productInfo.IPCountText;
productInfotext += Environment.NewLine;

productInfotext += "License State:" + productInfo.LicenseState;
productInfotext += Environment.NewLine;

productInfotext += "Serial:" + productInfo.Serial;
productInfotext += Environment.NewLine;

productInfotext += "Expiration:" + productInfo.LicenseExpiration;
productInfotext += Environment.NewLine;

public class ProductInfo
{
    public Version EngineVersion { get; set; }
    public Version AuditVersion { get; set; }
    public int IPCount { get; set; }
    public string IPCountText { get; set; }
    public string Serial { get; set; }
    public DateTime LicenseExpiration { get; set; }
    public LicenseState LicenseState { get; set; }
}

public enum LicenseState
{
    NotLicensed = 0,
    Licensed,
    Unknown
}
```

### Get Engine Information:

```
ScanManagementServiceClient client = getClient();
RetinaEngineStatus retinaEngineStatus = client.GetEngineInformation();

public enum RetinaEngineStatus
{
    UnknownError,
    Stopped,
    StartPending,
    StopPending,
    Running,
    ContinuePending,
    PausePending,
    Paused,
    Restricted,
    Unknown = -1,
}
```

### Get Package Support:

```
ScanManagementServiceClient client = getClient();
PackageFileInfo package;

// get package support for all jobs
package = client.GetPackageSupportForAllJobs();
// get package support for specific job
package = client.GetPackageSupportForJob("12345", "abcde");

string filePath = Path.Combine(@"c:\tmp", package.FileName);

try
{
    using (var fileStream = new FileStream(filePath, FileMode.CreateNew))
    {
        fileStream.Seek(0, SeekOrigin.Begin);
        fileStream.Write(package.Package, 0, package.Package.Length);
    }
}
```

### Update Retina:

```
ScanManagementServiceClient client = getClient();
client.UpdateRetina();
```