



# BeyondTrust

**Network Security Scanner  
Scan Management Service Guide**  
*Powered By Retina*

## Table of Contents

---

<b>Introduction</b> .....	<b>3</b>
<b>Use the Scan Management Service</b> .....	<b>4</b>

## Introduction

BeyondTrust 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 Network Security Scanner API and want to manage scans remotely.

# Use the Scan Management Service

## Overview

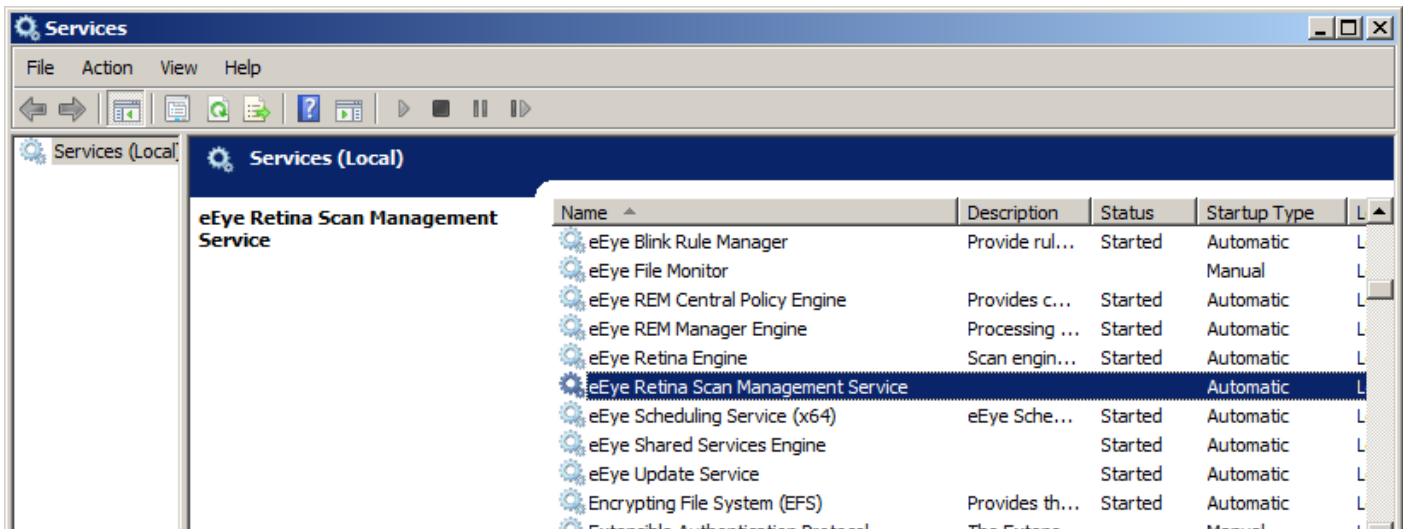
The WCF (Windows Communication Foundation) Scan Management Service enables you to remotely manage scans from any application that can send and consume a SOAP message. The WCF Network Security Scanner Scan Management service provides a subset of features that are available in the Network Security Scanner.

## Security

Scan Management Service supports Windows-based authentication and certificate-based authentication.

## Configuration

The Scan Management Service is a self-hosting WCF Windows service. The service is installed with the Network Security Scanner:



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.Servi ce.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"

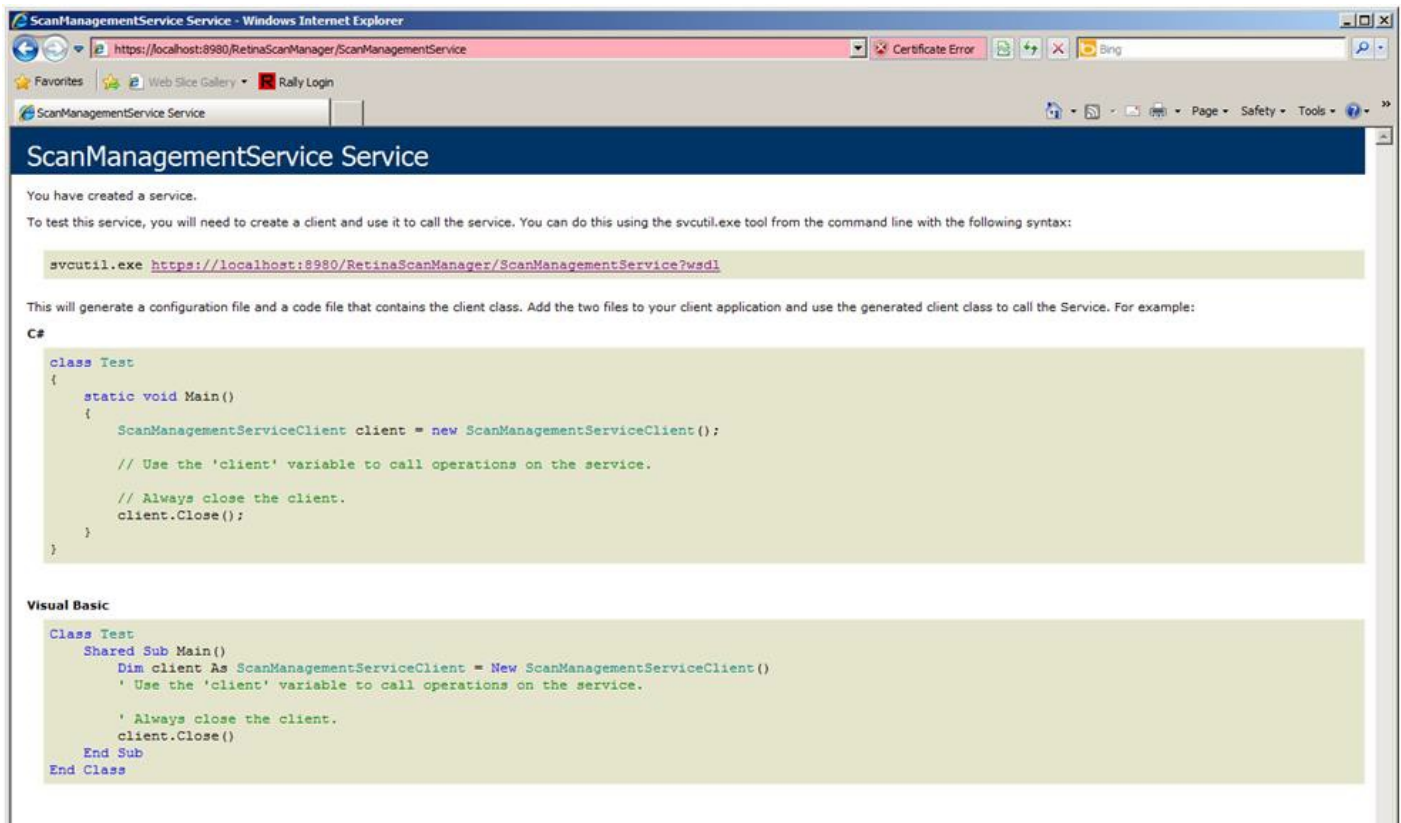
disableservice         Disable the Retina management service

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
Bing
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 svcutl.exe tool from the command line with the following syntax:

svcutl.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 at

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

## Class Reference

**Namespace:** Network Security Scanner.Service.ScanManager

### Methods:

Name	Description
Clean (UInt32 flags)	Cleans the Network Security Scanner Queue, Schedule, Logs, and Jobs
DeleteCredential (string description)	Removes credentials
GetAuditInformation (int rthId)	Returns Network Security Scanner audit information based on retina RTH ID
GetEngineInformation ()	Returns Network Security Scanner engine information
GetJobStatus (string scanID);	Returns Network Security Scanner job statuses based on Scan ID
GetJobStatusAll ()	Lists the Network Security Scanner jobs
GetPackageSupportForAllJobs ()	Runs the Package Support Tool to collect log files and downloads result packages
GetPackageSupportForJob (stringjobID, string groupID)	Runs the Package Support Tool to collect log files for a specific scan job and downloads result packages
GetProductInformation ()	Returns Network Security Scanner product information
GetScheduledJobs ()	Lists scheduled Network Security Scanner jobs
PauseScan (string scanID)	Pauses a running scan based on Scan ID
ResumeScan (string scanID)	Resumes a paused scan based on ScanID
SetCredential (stringusername, string password, string description)	Sets credential to be used for Network Security Scanner Scans
StartScan (stringscanID)	Starts a scan based on ScanID
StopScan (string scanID)	Stops a scan based on ScanID
StopSchedule (string scanID)	Removes scheduled scan
UpdateConfiguration (ScanFileInfo fileInfo)	Updates Network Security Scanner configuration information
UpdateRetina ()	Downloads the latest versions and updates of the Network Security Scanner
UpdateLicense (stringserial)	Updates Network Security Scanner serial information
RegistrationInfo (registrationInfo)	Updates Network Security Scanner license registration information
UploadAddress (ScanFileInfo fileInfo)	Uploads an Addressgroup file to be used for Network Security Scanner
UploadPort (ScanFileInfo fileInfo)	Uploads a Port group file to be used for Network Security Scanner
UploadAudits (ScanFileInfo fileInfo)	Uploads an Audit group file to be used for Network Security Scanner
UploadScanRequest (ScanFileInfo fileInfo)	Uploads a Scan Request file to be used for Network Security Scanner
UploadJob (ScanFileInfo fileInfo)	Uploads a Job file to be used for Network Security Scanner

## Code Examples

### 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` error message is provided 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 the Network Security Scanner:**

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