Administrator's guide

Protectimus 2FA Platform Integration with Citrix XenApp

Ver. 1.0.1 EN
1 May 2017

Contents

Preface

Authentication based on only a single factor — passwords — cannot be considered sufficiently reliable for systems with elevated security requirements. The simultaneous use of several factors in an authentication system significantly increases the level of protection against unauthorized access.

Many companies offer their own implementations of multiple-factor authentication, but the problem with them is that they're bureaucratic monsters in every way: they have poor communication and customer service; their product is bloated, awkward, and difficult to use; there is a lack of up-to-date information, and it's impossible to get any information when you need it most. On top of that, they operate monopolistically, sending their clients enormous invoices after agreements have already been signed, without having published their prices publicly.

Protectimus's goal is to offer a better two-factor authentication solution in terms of price and ease of use, while providing a high level of quality and system reliability.

Solution overview

The product we offer will allow:

- **Clients to** build more secure services on top of it, easily and accessibly regardless of the size of the company.
- **Clients' users** to reliably protect their accounts from unauthorized access

Protectimus offers a complex solution that includes not only an authentication system, but also a wide range of software- and hardware-based tokens, as well as other means of delivery of one-time passwords (OTPs).

The authentication system can function as a standalone program, installed on a client's servers, or as a cloud-based service, providing a SaaS solution. Protectimus has already ensured the stable, problem-free operation of your systems.

Protectimus solves the problem of two-factor authentication on all levels. Each client can find a solution that best meets their demands.

System features

The Protectimus platform supports a wide range of operating systems (from Linux and FreeBSD to any version of Windows).

The system supports both current and past versions of popular browsers: Google Chrome, Mozilla Firefox, and Internet Explorer.

All the system's components support existing software development standards, as well as OATH standards for OTP authentication, thanks to the ability to use third-party tokens and competitors' tokens in Protectimus.

Protectimus provides functionality to multiple unconnected copies of the software in different geographical areas near the client's users. It also provides the possibility to work with several nodes for clients whose users are located in different parts of the world.

Algorithms used to generate one-time passwords

For generating one-time passwords, the following algorithms are used:

- HMAC – hash-based message authentication code: RFC2104
- HOTP – hash-based one-time password: RFC4226
- TOTP – time-based one-time password: RFC6238
- OCRA – OATH Challenge-Response Algorithms: RFC6287

These algorithms were developed by the Initiative for Open Authentication (OATH), which aims to standardize authentication methods. These tried and tested algorithms have become de facto standards for two-factor authentication.

Protectimus Solutions LLP is a coordinating member of the OATH initiative. This solution is certified to be consistent with the above standards.

Citrix Ready Partner status

The Protectimus two-factor authentication platform is integrated with Citrix NetScaler Gateway and is a Citrix Ready Partner.
The Citrix Ready program is designed to certify the compatibility of third-party software solutions with Citrix products. That way, when choosing additional software for their systems, Citrix users can be sure of the reliability and full compatibility of these solutions with their systems.

The Protectimus two-factor authentication solution has demonstrated compatibility with such products as Citrix NetScaler Gateway 10.1, NetScaler Gateway 10.5, and NetScaler Gateway 11.0. Citrix Access Gateway software is used for secure remote access to key applications and data, and the implementation of fine-grained control over these applications.

Tech stack

#	Tools	Name (Version)
1	Java	7
2	Web/App Server	Tomcat 7.0
3	Framework	Spring 3.1.0, Apache Tapestry 5.3.7
4	GUI	Twitter Bootstrap, JQuery
5	ORM	Spring JDBC
6	Database	PostgreSQL 9.3
7	Building	Maven 3
8	High-performance, distributed memory object caching system	Memcached
9	Application Load Balancing and Content Caching	Nginx

Qualities and characteristics of the software

In the development of the software, the following have been used:

- Standard mechanisms and libraries
- Java Programming Style Guidelines (Java™ Coding Style Guide)
- DRY (Don't Repeat Yourself) and DIE (Duplication Is Evil) principles.
- Testing during development (TDD)

Solution architecture

Protectimus is built on SOA, MVC, and RESTful principles, and other core practices. Consider the solution's overall architecture, shown in Figure 1.

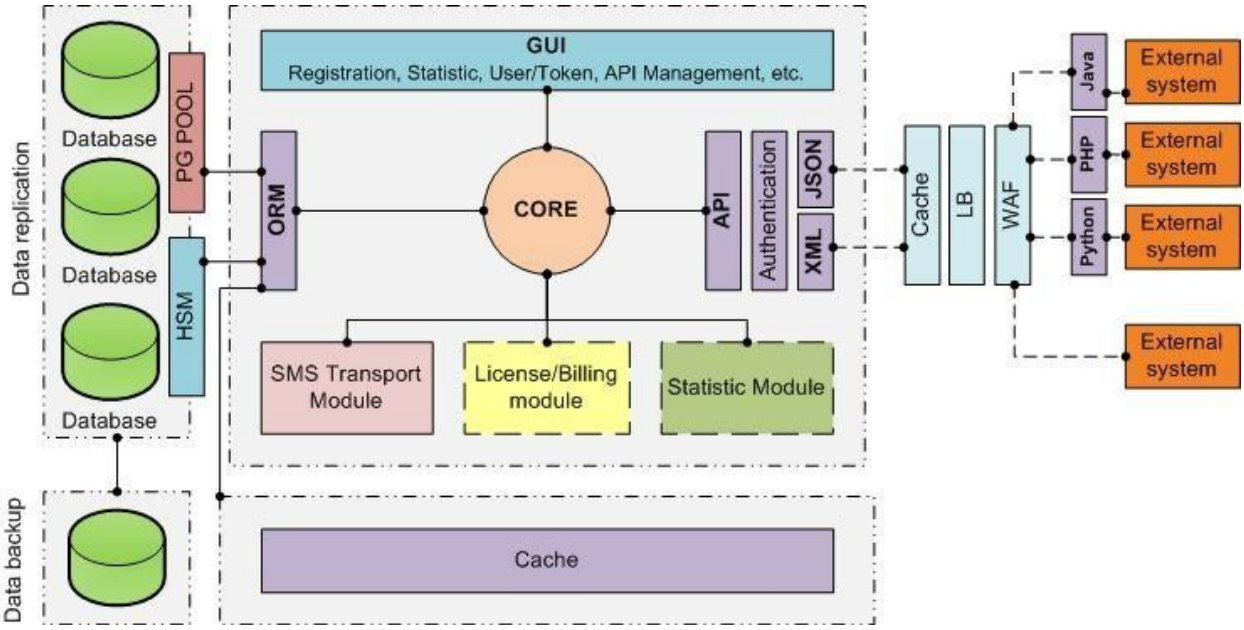


Figure 1. Overall architecture of the solution (possible delivery option)

The use of backup copies and duplication of data provides increased system sustainability and data protection.

To increase performance, optimization is applied on numerous levels:

- client-side: caching at the browser level, minimization of resources (HTML, CSS, and JavaScript), optimization of images, CSS sprites, etc.;
- server-side: caching (Memcached), optimization of static content processing (Nginx), optimization of application server configuration, load balancing, etc.;
- database-side: optimization of database configuration, use of indexing, partitioning.

System management is implemented through a graphical user interface, based on Twitter Bootstrap, which provides an adequate content display of content in different browsers and on different devices.

Interaction between Protectimus and client systems is implemented through a RESTful API, with data transmitted as XML or JSON. To ease processes of integration, libraries have been developed in Java, Python, and PHP. API documentation is available on the Protectimus site under "Materials".

The certified authentication model allows the use of any tokens developed according to standard OATH algorithms (HOTP, TOTP, and OCRA) in Protectimus. Also, the SMS delivery module provides one-time passwords to those who do not wish or cannot use other types of tokens.

Integration process / deployment of software solution

Interaction of Protectimus with NetScaler gateway is by means of the RADIUS protocol.

Solution integration consists of setting up and configuring Protectimus RProxy, as well as configuring NetScaler Gateway authentication policies.

Configuring authentication policies in NetScaler allows the transmission of an authentication request over the RADIUS protocol, which is then received and processed by the Protectimus RProxy component. Having received the request, the RProxy component, in turn, contacts the Protectimus authentication server to verify the one-time password supplied by the user.

There are several methods of delivery of the Protectimus authentication server. To expedite the process of establishing reliable authentication, provide free trials, and just to put the figurative burden on "other people's shoulders", we've designed a service using the SaaS model.

Installation of the Protectimus platform on your own hardware is a second option. This option allows you to implement authentication in an isolated environment.

If necessary, Protectimus specialists can also prepare an individualized cluster in the cloud, according to the client's needs.

The functionality of the system is preserved, regardless of which server placement option you choose. To switch from one option to another, all that's needed is to change a few settings, connecting to the API at its new address.

Next, a more detailed overview of the integration process.

Protectimus authentication server

To begin using the Protectimus SaaS service, simply register at the following address: https://service.protectimus.com

To install and use the Protectimus platform on your own hardware, a little more work is required. Let's examine this process in greater detail.

Deployment of the Protectimus platform

The authentication platform is available upon request from Protectimus support, at the following e-mail address: support@protectimus.com.

The authentication server is written in Java and is cross-platform.

Before installing the authentication platform on your server, Java (JDK version 7) must be installed, as well as the PostgreSQL DBMS, version 9.2 or later.

In PostgreSQL, a new database must be created for use by the platform.

After receiving the archive containing the Protectimus platform files, expand it in any directory (PLATFORM_DIR) and run the **install_platform.sh** or **install_platform.bat** script, depending on your chosen operating system. The script will ask you questions regarding the database connection, and required tables will be created in the database. The script will take some time to run.

If the script finishes without errors, in the PLATFORM_DIR directory, a configuration file called protectimus.platform.properties will be created, in which the following settings are configured:

```
jdbc.username - database username
jdbc.password - database password
jdbc.driverClassName - postgresql database driver
name ( org.postgresql.Driver )
jdbc.url - database connection info
```

You'll also need to add the following settings to this file:

```
default.from.address - address from which email will be sent
smtp.host - SMTP host smtp.port - SMTP port
smtp.user - SMTP username smtp.password - SMTP password
defaultEncoding - default character encoding (UTF-8)
files.directory - directory in which platform files are stored
cache.enabled - enables or disables caching Allowed values are
true or false. If set to true, the platform will use the built-in
caching mechanism. licence.file.path - path to the license file;
when using the platform in demo mode, this parameter should be
left empty (licence.file.path=)
```

SMPP server connection configuration values for sending SMS messages are also set in this file.

```
# SMPP server connection configuration, if
used. The parameter names speak for themselves. sms.use.smpp=true
smpp.server.login smpp.server.password smpp.server.host
smpp.server.port smpp.from.address

# Number of threads to use for sending SMS messages
smpp.sending.task.thread.pool.size=10 smpp.reconnect.interval=5000
smpp.pdu.processors.thread.pool=3 smpp.watch.traffic=true
smpp.watch.traffic.period.seconds=60

Additional parameters correspond exactly to the SMPP 3.4
specification.
smpp.dest.addr.npi=0x01 smpp.gateway.addr.npi=0x01
smpp.source.addr.npi=0x01 smpp.source.addr.ton=0x05
smpp.gateway.addr.ton=0x00
```

```
smpp.dest.addr.ton=0x01
smpp.protocol.id=0
smpp.replace.msg.if.present=1
smpp.data.coding=0x05
smpp.default.msg.id=0
smpp.service.type=CMT smpp.system.type=
smpp.enquire.link.timer=20000
smpp.transaction.timer=10000
```

After configuring these parameters, the platform is ready to be launched. The following options are available:

1) For running in test mode on Jetty, run the start.sh or start.bat script from the PLATFORM_DIR folder. The server will be started on port 8080, and the platform will be available from the address http://localhost:8080
2) Running from a Tomcat servlet container. For this:
   download and install Tomcat according to the instructions on the official site;
   copy the file **multipass-platform.war** from the PLATFORM_DIR folder to TOMCAT_HOME/webapps
   copy the configuration file **protectimus.platform.properties** to the TOMCAT_HOME/conf folder.
   launch Tomcat according to the instructions on the official site.

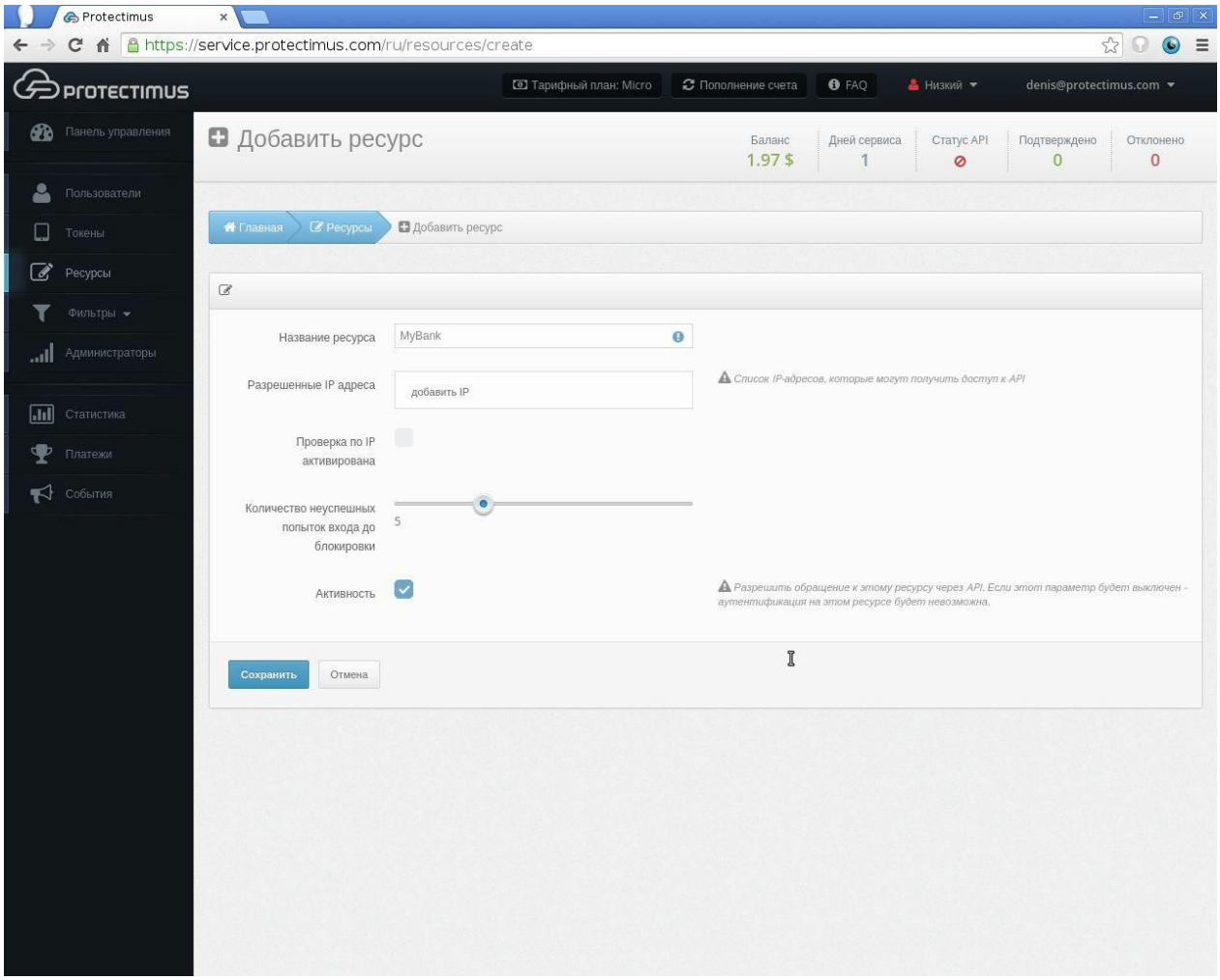
After launching the platform, you'll need to register in the system. Additionally, you'll need to get a license. To do so, go to http://platform_path/licensing, select the option you require, and get the license key. Using the key you received, you can pay for and download your license online, at https://service.protectimus.com/platform/licensing. If you require an alternate payment method, contact Protectimus customer service.

After receiving the license file, download it to the server and provide the path to the license file in the **licence.file.path** parameter, in the file named **protectimus.platform.properties**.

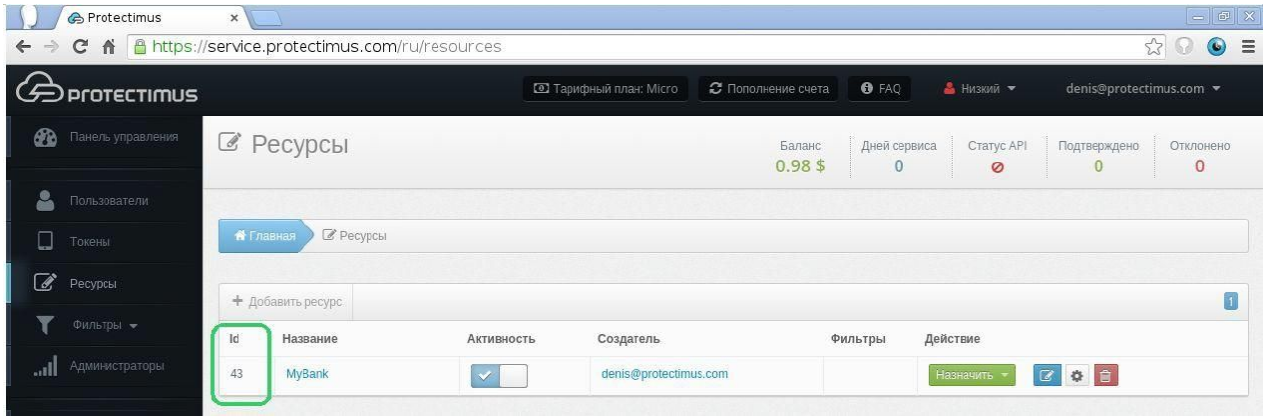
Working with the Protectimus system

Resources

Resources are used to logically group users and tokens, and to easily manage them. To create a resource, click the "Resources" button in the menu on the left, and then click the "Add resource" button in the table header. This will take you to the resource adding page, where you'll need to specify just a name for the resource and other parameters, if desired.

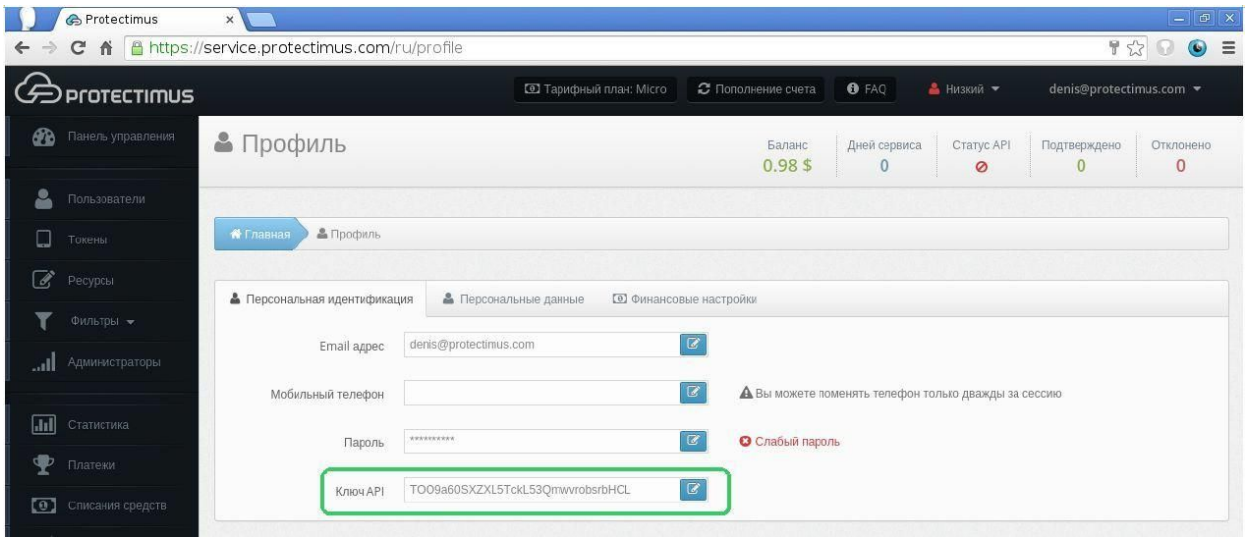


After creating the resource, you'll be taken to a page with a list of available resources, where you can see the resource you've just created. In addition, the ID of the resource will be displayed in the table; you'll use it in Protectimus's connection settings.



API key

To connect to Protectimus, you'll also need an API key, which is located in the user profile. In order to access a user profile, click the user's login in the top right corner of the interface, and choose the "Profile" entry from the drop-down list.

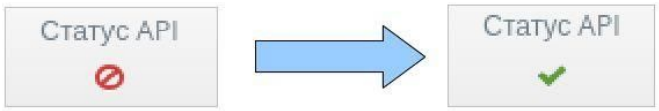


API activation

When using the SaaS solution, you'll need to activate a payment plan for the API. To do so, navigate to the "Payment plants" page at http://service.protectimus.com/pricing and activate the plan you'd like to use. Your account won't be charged until you activate a plan, but you won't be able to use the API until you do so. You can also deactivate a payment plan

at any moment, if for some reason you won't need to use the service for more than one day. When you deactivate a plan, a one-time fee is charged to your account for that day, according to the rates in the active plan. When a plan is active, you'll be charged once per day automatically.

After activating a payment plan, the API status icon will change to the "enabled" state, indicating that the service is ready for operation through the API.



The Protectimus system is ready for use. SMS and e-mail user authentication tokens will automatically be created when a user logs in through NetScaler for the first time. Other kinds of tokens can also be used after creating them on the Protectimus server. To receive additional information about the use of other kinds of tokens with NetScaler, contact Protectimus customer service.

Protectimus RProxy configuration

To receive the latest version of Protectimus RProxy, contact Protectimus customer service at support@protectimus.com.

For RProxy to function, Java 7 must be installed. RProxy can be started using

the following command: `java -jar RProxy.jar`

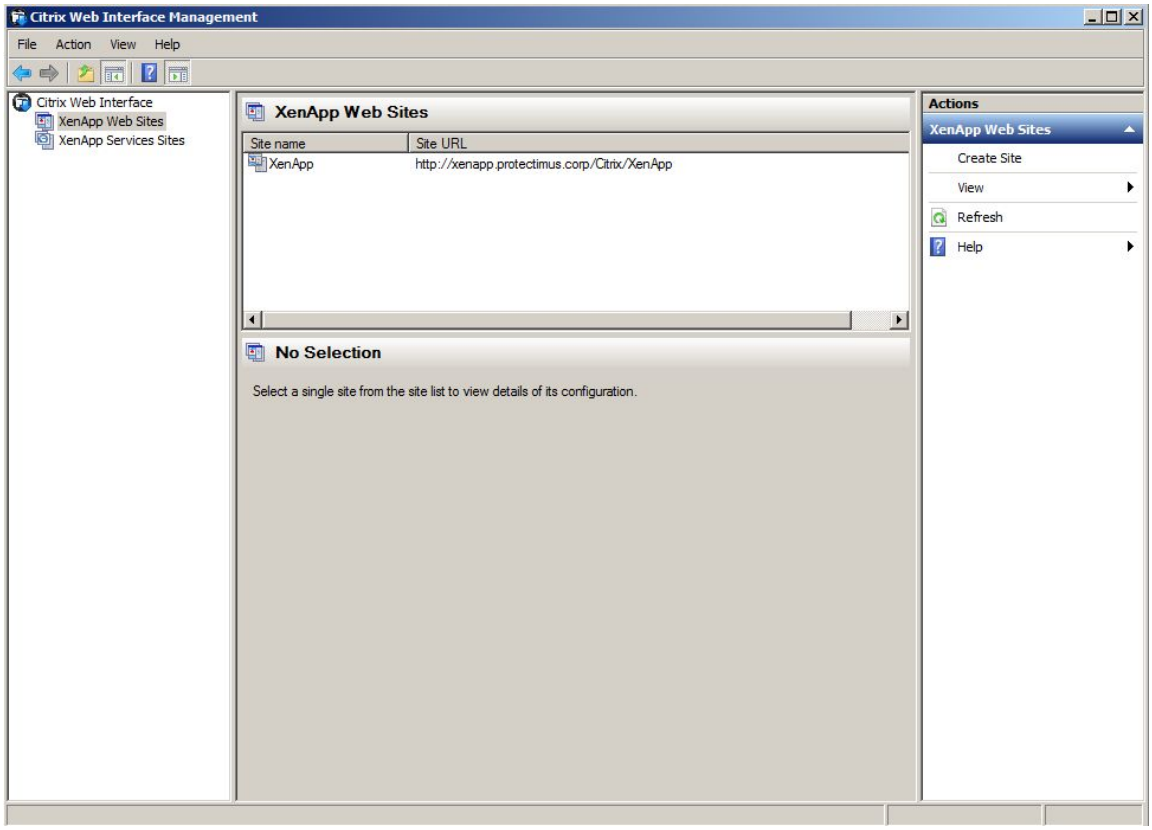
RProxy settings can be configured by specifying them in the **rproxy.properties** file, which must be located in the same directory as the executable. The standard configuration is as follows:

```
# RADIUS Server Settings rproxy.radius.port=1812
rproxy.radius.secret=[your_radius_secret]

# Protectimus API Settings
protectimus.login=[your_login@example.com]
protectimus.api.key=[your_API_key]

protectimus.api.url=https://api.protectimus.com
# If you are using the platform, the API URL will be something
like:
# protectimus.api.url= http://127.0.0.1:8080/

protectimus.resource.id=[id_of_the_resource]

# LDAP Settings
ldap.admin.name = CN=admin,CN=Users,DC=protectimus,DC=office
ldap.admin.password=[your_password]
ldap.url = ldap://[ip_ldap_server]:[port_ldap_server]
# e.g. 192.168.1.240:389
ldap.searchbase = DC=protectimus,DC=office auth.by.mail.group =
[mailgroup1,mailgroup2] auth.by.sms.group = [smsgroup]
auth.by.smart.group=[get_OTP_with_smartphone_group]

# If true - users who are not included in any of the above groups
# (without 2fa) will be rejected. The 'false' value allows these
# users to get in using only first factor.
restrict.access.for.not.specified.groups=false
```

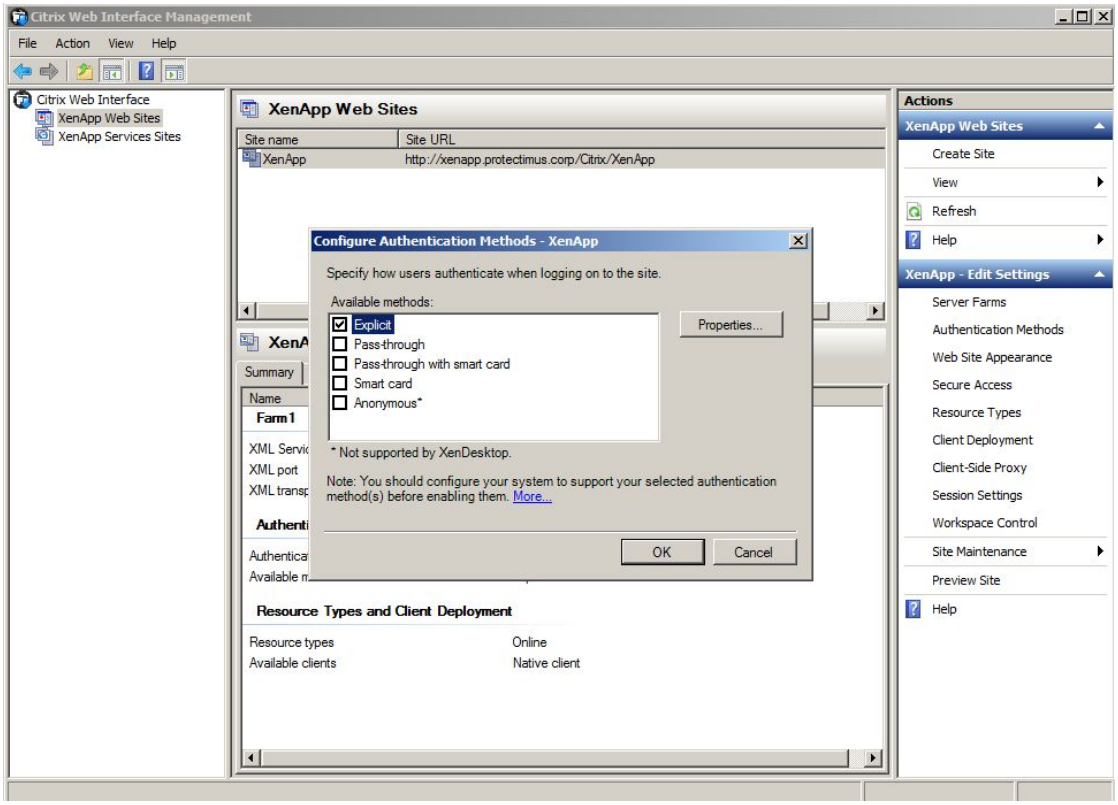
XenApp configuration

Next, you must create XenApp authentication policies.

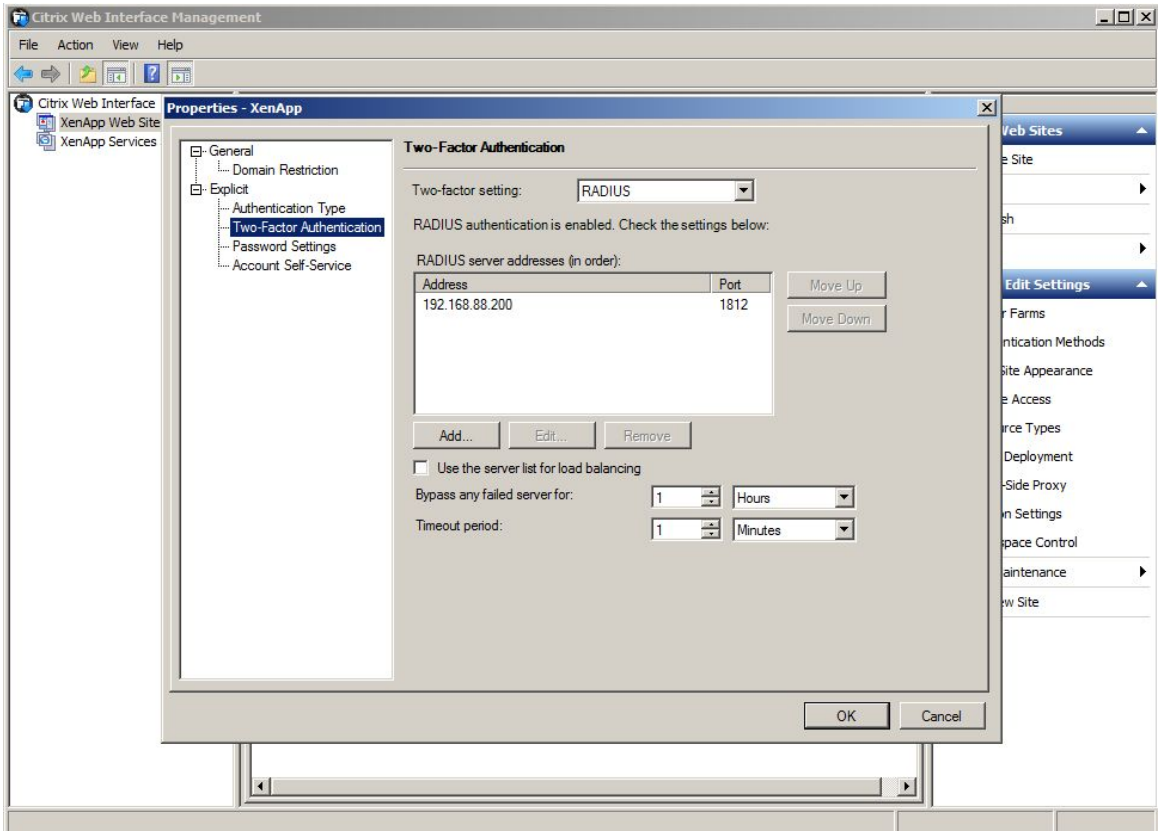
1) Navigate to Citrix Web Interface Management -> Web Sites, then right-click the site and click the “Authentication methods” button



2) Click the “Properties” button...



3) Navigate to the TWO-Factor Authentication -> Two-Factor settings -> RADIUS section



Click the “Add” button

Specify the server address and RProxy port

4) In the folder C:\inetpub\wwwroot\Citrix\sitepath\ in the web.config file

Find the parameter RADIUS_NAS_IP_ADDRESS and set the value of the RProxy address

5) In the folder C:\inetpub\wwwroot\Citrix\sitepath\conf create a file radius_secret.txt with the password specified in rproxy.properties in the parameter rproxy.radius.secret

6) To hide the OTP input field on the first step of authentication, do the following:

In the file C:\inetpub\wwwroot\Citrix\sitepath\app_data\include\loginMainForm.inc

Replace the line #255

```
<label id='lblPasscode' for='<%=Constansts.ID_PASSCODE%>'
```

by

```
<label style='display:none' id='lblPasscode'
for='<%=Constansts.ID_PASSCODE%>'
```

Replace the line #264

```
<input type='password' name='<%=Constants.ID_PASSCODE%>'
id='<%=Constants.ID_PASSCODE%>'
```

by

```
<input type='hidden' value='xenapp' name='<%=Constants.ID_PASSCODE%>'
id='<%=Constants.ID_PASSCODE%>'
```

Integration is now complete. If you have other questions, contact Protectimus customer service.

How to install system updates

To update the system, request a new version of the software. Afterwards, to update the platform, replace the WAR archive in the TOMCAT_HOME/webapps folder with the one you received, if working with a servlet container; or the one in PLATFORM_DIR if working with Jetty. After replacing the WAR archive, restart the application server.

To update RProxy, simply request a new version, as before. Replace the existing JAR archive with the new one you receive. Restart.

Our services

For problems, questions, and feedback:

support@protectimus.com

For partnerships and sales: sales@protectimus.com

Company information

Protectimus Solutions LLP

Phone:

United Kingdom: +44 20 3808 7124

Ukraine: +38 057 706 21 24

Russia: +7 499 677 16 34