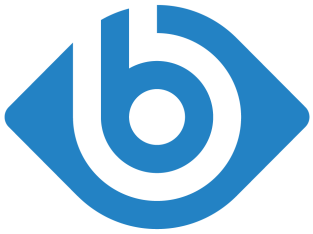
Configuring Balabit Shell Control Box for Blindspotter

May 05, 2017



BALABIT

CONTEXTUAL SECURITY INTELLIGENCE

Table of Contents

1. Configuring Balabit Shell Control Box for Blindspotter

To be able to add Balabit Shell Control Box as a data source to Blindspotter, you will have to configure Balabit Shell Control Box in a way so that it contains data that can be used by Blindspotter later.

Shell Control Box has the following requirements:

Type	Requirement
SCB version	Any supported version from version 4.0.4 onward, ideally the latest one
API	Enabled RPC API access on SCB
Access rights	A user account with search access rights
Data	Data that contains real, unique usernames linked to users other than root/administrator or a shared account. Ensure that there is least 2-3 months worth of data available in SCB, with at least 50 sessions per user.

Table 1. Shell Control Box prerequisites

To enable Blindspotter to fetch data from SCB, you will need SCB's IP address and the credentials:

- SCB's IP address
- SCB username
- SCB password

To be able to fetch activities, make sure that the connections include user names. To check this, navigate to **Search > Search** and check whether the **Username** column contains data.

This is important, because users will be fetched directly from SCB, and the activities performed will be linked to these users.

All questions, comments or inquiries should be directed to <info@balabit.com> or by post to the following address: Balabit SA 1117 Budapest, Alíz Str. 2 Phone: +36 1 398 6700 Fax: +36 1 208 0875 Web: *https://www.balabit.com/*

2. The SCB RPC API

Version 3 F3 and later of Balabit Shell Control Box can be accessed using a Remote-Procedure Call Application Programming Interface (RPC API).

The SCB RPC API allows you to access, query, and manage SCB from remote applications. You can access the API using the Simple Object Access Protocol (SOAP) protocol over HTTPS, meaning that you can use any programming language that has access to a SOAP client to integrate SCB to your environment. You can download simple, proof-of-concept clients for Python and other languages from the SCB web interface.

Accessing SCB with the RPC API offers several advantages:

- Integration into custom applications and environments
- Flexible, dynamic search queries and management

2.1. Requirements for using the RPC API

To access SCB using the RPC API, the following requirements must be met:

- Accessing the appliance via the RPC API must be enabled on the web interface. For details, see *Procedure 2.5, Enabling RPC API access to SCB (p. 6)*.
- The appliance can be accessed using the SOAP protocol over authenticated HTTPS connections. The WSDL describing the available services is available at *https://<ip-address-of-SCB>/rpc.php/<techversion>?wsdl*. For details on the client libraries tested with SCB see *Section 2.2, RPC client requirements (p. 4)*.
- The user account used to access SCB via RPC must have **read and write/perform** rights for the **Access RPC API** privilege. This is required for every type of RPC access, even for read-only operations. Members of the `api` group automatically have this privilege. For details on managing user privileges, see *Procedure 5.7.2, Modifying group privileges* in *The Balabit Shell Control Box 5 LTS Administrator Guide*.

Warning
Each SCB release provides a separate API with a new API version number. You are recommended to use the SCB version 5 LTS with the corresponding API version. Earlier versions are not supported

2.2. RPC client requirements

The client application used to access SCB must meet the following criteria:

- Support SOAP version 1.1 or later.
- Support WSDL version 1.1.
- Properly handle complex object types.
- Include a JSON decoder for interpreting the results of search operations.

The following client libraries have been tested with SCB.

Client name	Programming language	Status	Comments
Apache Axis 1	Java	Working	
Built-in .NET library	.NET	Working	SCB does not support the Expect HTTP Header feature, and must be disabled, for example, using `System.Net.ServicePointManager.Expect100Continue = false;`
Scio	Python	Partially working	Does not handle complex object types, so it cannot perform search queries.
SOAP::Lite	Perl	Working	■ Simple types can be used with the following format: `$service->$method(@params)` ■ Complex types work only with the following format: `$service->call($method, @params)` ■ Calls using the `$service->call()` format seem to work after doing at least one `$service->$method(@params)` call, for example, a login.
SOAP::WSDL	Perl	Not working	
Suds	Python	Working	

Table 2. SOAP libraries tested with SCB

2.3. Locking SCB configuration from the RPC API

Accessing SCB using the RPC API locks certain components of SCB from other users, just like accessing SCB using the web interface or the console. Locking SCB via RPC can be performed either explicitly by calling the `lockAcquire` function, or implicitly when an operation requires the lock. In either case, ensure that your

application verifies that the lock is received and properly handles if the component is locked by someone else (for example, because a user is accessing the component from the web interface).

For details on how locking works in SCB, see *Section 4.2.2, Multiple users and locking* in *The Balabit Shell Control Box 5 LTS Administrator Guide*.

2.4. Documentation of the RPC API

The documentation of the SCB RPC API is available online from the SCB web interface: select **Basic Settings > Management > RPC API settings > Open documentation**, or directly from the following URL: *https://<ip-address-of-SCB>/rpc-api-doc/*. This documentation contains the detailed description of the available services and classes.

2.5. Procedure – Enabling RPC API access to SCB

Purpose:

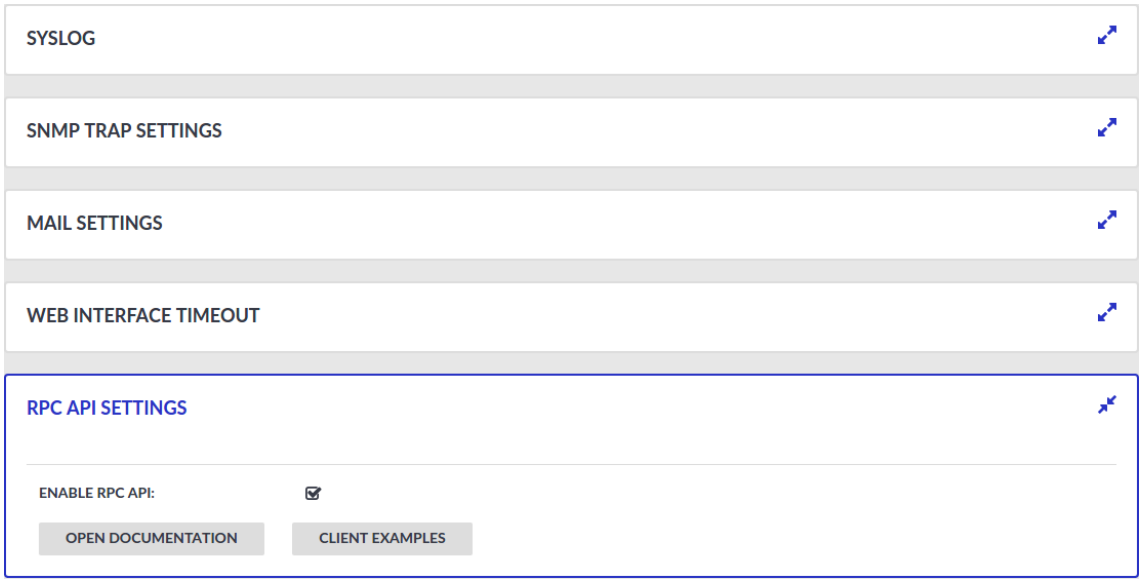
To configure SCB to accept RPC API connections, complete the following steps.

Steps:

Step 1. Login to the SCB web interface.

Step 2. Select **Basic Settings > Management > RPC API settings > Enable RPC API**.

*Figure 1. **Basic Settings > Management > RPC API settings** — Enabling RPC API access to SCB*



Step 3. Click COMMIT.

Expected result:

Users accounts belonging to a usergroup that have `read and write/perform` rights to the `Access RPC API` privilege can access SCB via the RPC API.

3. Usernames in RDP connections

When processing RDP connections, SCB attempts to extract the username from the connection. For example, you need the username to:

- Use gateway authentication for the connection. For details on gateway authentication, see *Section 18.2, Configuring gateway authentication* in *The Balabit Shell Control Box 5 LTS Administrator Guide*.
- Use usermapping policies. In this case, SCB compares the username on the server with the username on the gateway. For details on usermapping policies and gateway authentication, see *Procedure 18.1, Configuring usermapping policies* in *The Balabit Shell Control Box 5 LTS Administrator Guide* and *Section 18.2, Configuring gateway authentication* in *The Balabit Shell Control Box 5 LTS Administrator Guide*, respectively.

Note
In certain cases, SCB receives an empty username from the server, and the connection will be denied by the usermapping policy unless a policy is set for the connection that allows every user for the given group. To add such a policy, specify `*` in the **Username on the server** field of the usermapping policy. For a list of cases when SCB receives empty username, see *Section Windows settings that interfere with username extraction (p. 7)*.

- Search or filter connections by the username on the SCB search interface, or create automatic statistics based on the username.

SCB can record the username automatically in the following situations if the RDP connection is using Network Level Authentication (CredSSP), and usually in other scenarios as well. The known scenarios that interfere with RDP usernames are listed in *Section Windows settings that interfere with username extraction (p. 7)*.

To ensure that your users can access the target servers only when their username is recorded, you can configure SCB to terminate RDP connections if it cannot reliably extract the username. To terminate such connections, disable the **RDP Control > Settings > Permit unreliable usernames** option.

Windows settings that interfere with username extraction

The following settings on the Windows client or server can prevent SCB from correctly extracting the username from the RDP connection. As a result, the username is not visible on the **Search**, **Four Eyes** and **Active Connections** pages.

- The **DontDisplayLastUserName** option is enabled on the server. The **DontDisplayLastUserName** security setting of Windows servers specifies whether the username from the last successful login is displayed on the login screen as a default for the next login. To disable the **DontDisplayLastUserName** security setting, do one of the following.
  - Disable the `HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\System\DontDisplayLastUsername` registry setting. For more details, see the *DontDisplayLastUserName TechNet article*.

Note
Registry settings can be overridden by Group Policy settings.

- Disable this option in the `Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options` policy. For details, see *Do not display last user name in logon screen TechNet article*.

- There is no server-side authentication. To avoid this problem, ensure that your server requires authentication from the users.
- If the server is Windows 2003 Server or Windows XP and the **Allow to save credentials** or **Remember my credentials** options are enabled in the Remote Desktop client application. In this case, disable these options on the client, and delete any credentials that have already been saved on the client.