BOMGAR™

API Programmer's Guide 1.14.0 (PAM 15.3.x)

TC:5/25/2016

Table of Contents

Privileged Access Management API Programmer's Guide

Version 1.13.0 (for Bomgar 15.1.x and 15.2.x)

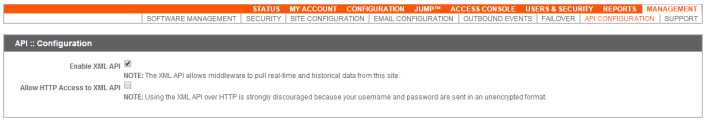
Version 1.14.0 (for Bomgar 15.3.x)

Front-end integration of the Bomgar API enables customers to correlate Bomgar sessions with third-party or in-house developed applications to pull report data, issue commands, or automatically save a backup of the Bomgar Appliance's software configuration on a recurring basis.

One common example of API integration would be linking a customer relationship management ticketing system to Bomgar sessions.

You could also add a feature to an application to enable the user to start a session from directly within that program instead of the Bomgar access console.

To use the Bomgar API, ensure that the **Enable XML API** option is checked on the **Management > API Configuration** page of the **/login** administrative interface.



For the examples in the following pages, a sample URL of **access.example.com** is used. Please replace this URL with your Bomgar Appliance's public site URL.

The command and reporting APIs return XML responses that declare a namespace. If you are parsing these responses with a namespace-aware parser, you will need to set the namespace appropriately or ignore the namespace while parsing the XML.

- Reporting API: https://www.bomgar.com/namespaces/API/reporting
- Command API: https://www.bomgar.com/namespaces/API/command

Note: *The above namespaces are returned XML data and are not functional URLs.*

Command API

The Bomgar command API is designed to send commands to your Bomgar Appliance from an outside application. Commands can get or set session attributes, join an existing session, or terminate a session. You also can check the health of your appliance or get information about your Bomgar API version.

Commands are executed by sending an HTTP request to the appliance. Send the request using any HTTPS-capable socket library or scripting language module, web browser, or URL fetcher such as **cURL** or **wget**. Use either **GET** or **POST** as the request method.

IMPORTANT!

When making consecutive API calls, you must close the connection after each API call.

***Note:** By default, access to the API is SSL-encrypted; however, you can choose to allow HTTP access by checking the **Allow HTTP Access to XML API** option on the **Management > API Configuration** page of the **/login** administrative interface. **It is highly recommended that HTTP remain disallowed as a security best practice**.*

The command API URL is **https://access.example.com/api/command**.

An XML schema describing the command API response format is available at **https://access.example.com/api/command.xsd**.

Required Parameters for Command API

username=[string]	The username to use to issue commands. For all commands except **get_api_info**, this user must have permission to use the command API and must be an admin.
password=[string]	The password associated with this username.
action=[string]	The type of action to perform. Can be **join_session**, **set_session_attributes**, **get_session_attributes**, **import_jump_shortcut**, **terminate_session**, **check_health**, or **get_api_info**.

The command API returns XML responses that declare a namespace. If you are parsing these responses with a namespace-aware parser, you need to set the namespace appropriately or ignore the namespace while parsing the XML.

- Command API: https://www.bomgar.com/namespaces/API/command

***Note:** The above namespace is returned XML data and is not a functional URL.*

API Command: set_session_attributes

The **set_session_attributes** command sets the external key and other custom attributes for an active session.

In order to issue the **set_session_attributes** command, you must supply the username and password for a Bomgar user account. That account must have the permission **Allowed to Use Command API** along with the permission **Administrator**.

Required Parameter for set_session_attributes

lsid=[string]	The ID of the session whose attributes you wish to set. The session must currently be active.

Optional Parameters for set_session_attributes

session.custom.external_key=[string]	An arbitrary string that can link this session to an identifier on an external system, such as a customer relationship management ticket ID. This has a maximum length of 1024 characters.
session.custom.[custom field]=[string]	The code name and value of any custom fields. These fields must first be configured in **/login > Management > API Configuration**. Each attribute must be specified as a different parameter. Each custom field has a maximum length of 1024 characters. The maximum total size of all combined custom fields, including the external key, must be limited to 10KB.

***Note:** If an attribute is not listed in the URL, it will keep its existing value. To clear an attribute, you must set the attribute to an empty string.*

XML Response for set_session_attributes Query

<success>	Returns a message of **Session attributes were set** if the attributes were set successfully.
<error>	Returns an error message if the attributes were not set successfully.

Query Examples: set_session_attributes

Set external key for session c69a8e10bea9428f816cfababe9815fe	https://access.example.com/api/command?username=test&password=test&action=set_session_attributes&lsid=c69a8e10bea9428f816cfababe9815fe&session.custom.external_key=ABC123
Set a custom value for session c69a8e10bea9428f816cfababe9815fe	https://access.example.com/api/command?username=test&password=test&action=set_session_attributes&lsid=c69a8e10bea9428f816cfababe9815fe&session.custom.custom_field1=Custom%20Value

API Command: get_session_attributes

The **get_session_attributes** command returns attributes set for an active session.

In order to issue the **get_session_attributes** command, you must supply the username and password for a Bomgar user account. That account must have the permission **Allowed to Use Command API** along with the permission **Administrator**.

Required Parameter for get_session_attributes

lsid=[string]	The ID of the session whose attributes you wish to get. The session must currently be active.

XML Response for get_session_attributes Query

<custom_attributes>	Contains a **<custom_attribute>** element for each custom attribute set for the session.
<error>	Returns an error message if the attributes were not retrieved successfully.

Element Names and Attributes

	/custom_attributes/custom_attribute
display_name (attribute)	The display name assigned to the custom attribute.
code_name (attribute)	The code name assigned to the custom attribute.

Query Example: get_session_attributes

Get custom attributes for session c69a8e10bea9428f816cfababe9815fe	https://access.example.com/api/command?username=test&password=test&action=get_session_attributes&lsid=c69a8e10bea9428f816cfababe9815fe

API Command: import_jump_shortcut

The **import_jump_shortcut** command creates a Jump shortcut. When dealing with a large number of Jump shortcuts, it may be easier to import them programmatically than to add them one by one in the access console.

In order to issue the **import_jump_shortcut** command, you must supply the username and password for a Bomgar user account. That account must have the permission **Allowed to Use Command API** along with the permission **Administrator**.

Required Parameters for import_jump_shortcut - Local Jump

local_jump_hostname=[string]	The hostname of the endpoint to be accessed by this Jump shortcut.
group=[string]	The code name of the team with which this Jump Item should be associated. ***Note:*** *Using the import method, a Jump Item cannot be associated with a personal Jump Group.*

Optional Parameters for import_jump_shortcut - Local Jump

tag=[string]	You can organize your Jump Items into categories by adding a tag. This string has a maximum of 1024 characters.
comments=[string]	You can add comments to your Jump Items. This string has a maximum of 1024 characters.
jump_policy=[string]	The code name of a Jump Policy. You can specify a Jump Policy to manage access to this Jump Item.
session_policy=[string]	The code name of a session policy. You can specify a session policy to manage the permissions available on this Jump Item.

Required Parameters for import_jump_shortcut - Remote Jump

remote_jump_hostname=[string]	The hostname of the endpoint to be accessed by this Jump shortcut.
jumpoint=[string]	The code name of the Jumpoint through which the endpoint is accessed.
group=[string]	The code name of the team with which this Jump Item should be associated. ***Note:*** *Using the import method, a Jump Item cannot be associated with a personal Jump Group.*

Optional Parameters for import_jump_shortcut - Remote Jump

tag=[string]	You can organize your Jump Items into categories by adding a tag. This string has a maximum of 1024 characters.

comments=[string]	You can add comments to your Jump Items. This string has a maximum of 1024 characters.
jump_policy=[string]	The code name of a Jump Policy. You can specify a Jump Policy to manage access to this Jump Item.
session_policy=[string]	The code name of a session policy. You can specify a session policy to manage the permissions available on this Jump Item.

Required Parameters for import_jump_shortcut - Remote Desktop Protocol

rdp_hostname=[string]	The hostname of the endpoint to be accessed by this Jump shortcut.
jumpoint=[string]	The code name of the Jumpoint through which the endpoint is accessed.
group=[string]	The code name of the team with which this Jump Item should be associated. ***Note:*** *Using the import method, a Jump Item cannot be associated with a personal Jump Group.*

Optional Parameters for import_jump_shortcut - Remote Desktop Protocol

rdp_username=[string]	The username to sign in as.
domain=[string]	The domain the endpoint is on.
display_size=[string]	The resolution at which to view the remote system. Can be **primary** (default - the size of your primary monitor), **all** (the size of all of your monitors combined), or **XxY** (where **X** and **Y** are a supported width and height combination - e.g., **640x480**).
quality=[string]	The quality at which to view the remote system. Can be **low** (2-bit gray scale for the lowest bandwidth consumption), **best_perf** (default - 8-bit color for fast performance), **perf_and_qual** (16-bit for medium quality image and performance), or **best_qual** (32-bit for the highest image resolution). This cannot be changed during the remote desktop protocol (RDP) session.
console=[boolean]	**1**: Starts a console session. **0**: Starts a new session (default).
ignore_untrusted=[boolean]	**1**: Ignores certificate warnings. **0**: Shows a warning if the server's certificate cannot be verified.
tag=[string]	You can organize your Jump Items into categories by adding a tag. This string has a maximum of 1024 characters.
comments=[string]	You can add comments to your Jump Items. This string has a maximum of 1024 characters.
jump_policy=[string]	The code name of a Jump Policy. You can specify a Jump Policy to manage access to this Jump Item.

session_policy=[string]	The code name of a session policy. You can specify a session policy to manage the permissions available on this Jump Item.

Required Parameters for import_jump_shortcut - Shell Jump Shortcut

shelljump_hostname=[string]	The hostname of the endpoint to be accessed by this Jump shortcut.
jumpoint=[string]	The code name of the Jumpoint through which the endpoint is accessed.
protocol=[string]	Can be either **ssh** or **telnet**.
group=[string]	The code name of the team with which this Jump Item should be associated. ***Note:*** *Using the import method, a Jump Item cannot be associated with a personal Jump Group.*

Optional Parameters for import_jump_shortcut - Shell Jump Shortcut

shelljump_username=[string]	The username to sign in as.
port=[integer]	A valid port number from **1** to **65535**. Defaults to **22** if the protocol is **ssh** or **23** if the protocol is **telnet**.
terminal=[string]	Can be either **xterm** (default) or **VT100**.
keep_alive=[integer]	The number of seconds between each packet sent to keep an idle session from ending. Can be any number from **0** to **300**. **0** disables keep-alive (default).
tag=[string]	You can organize your Jump Items into categories by adding a tag. This string has a maximum of 1024 characters.
comments=[string]	You can add comments to your Jump Items. This string has a maximum of 1024 characters.
jump_policy=[string]	The code name of a Jump Policy. You can specify a Jump Policy to manage access to this Jump Item.
session_policy=[string]	The code name of a session policy. You can specify a session policy to manage the permissions available on this Jump Item.

XML Response for import_jump_shortcut Query

<success>	Returns a message of **Successfully imported Jump Item shortcut** if the import succeeded.
<error>	Returns an error message if the import failed.

Query Examples: import_jump_shortcut

Import a Local Jump shortcut to the endpoint with hostname "ABCDEF02", pinning it to team "remote_access"	https://access.example.com/api/command?username=test&password=test&action=import_jump_shortcut&local_jump_hostname=ABCDEF02&group=remote_access
Import a Local Jump shortcut to the endpoint with hostname "ABCDEF02", pinning it to team "remote_access" and specifying its tag, comments, Jump Policy, and session policy	https://access.example.com/api/command?username=test&password=test&action=import_jump_shortcut&local_jump_hostname=ABCDEF02&group=remote_access&tag=Frequent%20Access&comments=Web%20server&jump_policy=Notify&session_policy=Servers
Import a Remote Jump shortcut to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London", and pinning it to team "remote_access"	https://access.example.com/api/command?username=test&password=test&action=import_jump_shortcut&remote_jump_hostname=ABCDEF02&jumpoint=London&group=remote_access
Import a Remote Desktop Protocol shortcut to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London", and pinning it to team "remote_access"	https://access.example.com/api/command?username=test&password=test&action=import_jump_shortcut&rdp_hostname=ABCDEF02&jumpoint=London&group=remote_access
Import a Remote Desktop Protocol shortcut to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London", and pinning it to team "remote_access". Set the username, domain, display size, and quality. Make it a console session, and ignore untrusted certificates.	https://access.example.com/api/command?username=test&password=test&action=import_jump_shortcut&rdp_hostname=ABCDEF02&jumpoint=London&group=remote_access&rdp_username=admin&domain=example&display_size=1280x720&quality=perf_and_qual&console=1&ignore_untrusted=1
Import a Shell Jump shortcut to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London" on SSH, and pinning it to team "remote_access"	https://access.example.com/api/command?username=test&password=test&action=import_jump_shortcut&shelljump_hostname=ABCDEF02&jumpoint=London&protocol=ssh&group=remote_access
Import a Shell Jump shortcut to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London" on SSH, and pinning it to team "remote_access". Set the username, port, and terminal type, and set the keep-alive time to two minutes.	https://access.example.com/api/command?username=test&password=test&action=import_jump_shortcut&shelljump_hostname=ABCDEF02&jumpoint=London&protocol=ssh&group=remote_access&shelljump_username=admin&port=25&terminal=vt100&keep_alive=120

API Command: terminate_session

The **terminate_session** command terminates a session that is in progress.

In order to issue the **terminate_session** command, you will need to supply the username and password for a Bomgar user account. That account must have the permission **Allowed to Use Command API** along with the permission **Administrator**.

Required Parameter for terminate_session

lsid=[string]	The unique ID representing the session you wish to terminate.

XML Response for terminate_session Query

<success>	Returns a message of **Successfully terminated** if the termination was successful.
<error>	Returns an error message if the termination was not successful.

Query Examples: terminate_session

Session da4b510978a541d49398e88c66e28475 terminated	https://access.example.com/api/command?username=username&password=password&action=terminate_session&lsid=da4b510978a541d49398e88c66e28475

API Command: check_health

The **check_health** command returns XML data containing information about the Bomgar Appliance.

In order to issue the **check_health** command, you will need to supply the username and password for a Bomgar user account. That account must have the permission **Allowed to Use Command API** along with the permission **Administrator**.

XML Response for check_health Query

<appliance>	The hostname of the appliance. Also contains an id attribute that contains the appliance's GUID.
<version>	The version number and build number of the Bomgar software running on the appliance.
<success>	Integer value (**1** or **0**) indicating if the health check of the appliance was successful.
<error_message>	Returns an error message if a problem is found. If no error is found, this element will not be returned.
<failover_role>	The role the appliance plays in the failover relationship. Can be one of **none** (if failover is not configured), **primary**, or **backup**.
<enabled_shared_ips>	Contains an **<ip>** element for each IP address which is shared between the primary and backup appliances. If no shared IP addresses are enabled or if failover is not configured, this element is not returned.
<last_data_sync_time>	The date and time at which the last data sync occurred between the primary and backup appliances. Data is returned in ISO 8601 format. Also contains a **ts** attribute which displays the data sync time as a UNIX timestamp (UTC). If failover is not configured, this element is not returned.
<last_data_sync_status>	Contains a string showing the status of the last data sync. If failover is not configured, this element is not returned.

Query Example: check_health

check_health	https://access.example.com/api/command?username=test&password=test&action=check_health

HTTP Status Check

In addition to using the API command above, you can use **https://access.example.com/check_health** to check the health of an appliance. This returns an HTTP status of 200 if the probe is successful and 500 (Server Error) if not. While you will see a simple human-readable message showing success or failure, no other data is exposed.

API Command: get_api_info

The **get_api_info request** returns XML data containing the current API version information.

XML Response for get_api_info Query

<api_version>	The software version of the current Bomgar API.
<timestamp>	The server's current timestamp at the time this report was pulled.
<permissions>	The permissions of the user account used to issue this command. The permissions shown are detailed below.
<user_id>	The numeric ID of the Bomgar user making this API call.

Element Names and Attributes

	/get_api_info/permissions/permission
perm_use_command_api	Integer value (**1** or **0**) indicating if the user has permission to use the command API.
perm_use_reporting_api	Integer value (**1** or **0**) indicating if the user has permission to use the reporting API.
perm_admin	Integer value (**1** or **0**) indicating if the user is an administrator.
perm_view_reports	Indicates if the user has permission to view reports. Can be one of the following: **none** — Cannot view any reports. **user_sessions** — Can view reports in which they were the primary user. **team_sessions** — Can view reports in which one of the user's teammates was the primary user. **all_sessions** — Can view all reports.
perm_view_sd_recordings	Integer value (**1** or **0**) indicating if the user has permission to view session recordings.

Query Example: get_api_info

get_api_info	https://access.example.com/api/command?username=test&password=test&action=get_api_info

Access Console Scripting and Client Scripting API

The Bomgar Access Console scripting feature is composed of three parts:

1. The Bomgar Access Console Script file format
2. Command line parameters for the access console
3. The Bomgar client scripting API

The Bomgar Access Console Script File

A Bomgar Console Script (BRCS) is a file that contains a sequence of commands to be executed by the Bomgar access console. The file extension is in the format "brcs-<companySiteName>" (Company Site Name is the name used to access your Bomgar site). During installation the Bomgar access console will use the OS to associate the access console with the BRCS file type. Therefore, users can double-click a BRCS file and have it automatically executed by the Bomgar access console.

BRCS files have the following format:

```
BRCS1.0
<command>
<command>
…
```

This is more formally expressed as:

```
brcs_file = header , newline , commands ;
header = "BRCS" , version ;
version = digit , "." , digit ;
commands = command { newline , command } ;
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;
newline = "\n" | "\r\n" ;
```

Note that script files can have a maximum of 10 commands.

Each command consists of a set of key-value pairs separated by ”&”. The key in each pair is separated from the value by ”=”. Keys and values use the percent-encoding algorithm described in RFC3986 section 2.1. This is commonly referred to as url-encoding or url-escaping. It is commonly seen in the address bar of web browsers to represent the parameters passed to a web server. Commands have the following format:

```
action=<action>&parameter1=value1&parameter2=value2...
```

This is more formally expressed as:

```
command = "action=", value, [ parameters ] ;
parameters = "&", parameter, [ parameters ] ;
parameter = url_encoded_string, "=", url_encoded_string ;
```

```
url_encoded_string = {* see RFC 3986 *} ;
```

Command Line Parameters for the Access Console

Two command line parameters exist in the access console to support BRCS:

```
run-script <BRCS command>

run-script-file <path to BRCS file>
```

These command line parameters allow users to implement BRCS login via the command line.

Different behaviors can be seen when running a script from the command line depending on the state of the access console:

- If the access console is not running, then attempting to run a script from the command line causes the access console to start the login dialog. After the user successfully logs in, the script is run.
- If the access console is already running but the user is not logged in, then the login dialog is shown. After the user logs in, the script is run.
- If the access console is already running and the user is already logged in, then attempting to run a script from the command line causes the existing instance of the access console to run the script.

Access console exit status:

- If an invalid script is given on the command line, then the access console will terminate with an exit status > 0.
- If a valid script is given on the command line, then the access console will terminate with an exit status of 0.

Examples:

```
bomgar-acc-x64.exe --run-script "action=start_jump_item_

session&client.hostname=ABCEF02&session.custom.external_key=123456789"

bomgar-acc-x64.exe --run-script-file my_script_file.brcs-beta60
```

The Bomgar Client Scripting API

The client scripting API enables you to generate a Bomgar Console Scripting (BRCS) file which allows you to send commands to the Bomgar access console from external applications.

Customers can use the client scripting API to generate BRCS files that can start a session with a specific Jump Item or simply to log into the access console.

The client scripting API URL is **https://access.example.com/api/client_script**.

This API accepts a client type (**rep**), an operation to perform (**generate**), a command to put in the script file, and a set of parameters to pass to the command. Here is an example of a valid Client Scripting API request:

```
https://access.example.com/api/client_script?type=rep&operation=generate&action=start_jump_item_

session&client.hostname=ABCDEFG02
```

The above request prompts the user to download a Bomgar access console script file. After downloading the script file, the user can run it using the access console. In this case, the script file will contain commands to start a session with the Jump Client whose hostname, comments, public IP, or private IP matches the search string "ABCDEF02".

Note: *By default, access to the API is SSL-encrypted; however, you can choose to allow HTTP access by checking the* ***Allow HTTP Access to XML API*** *option on the* ***Management > API Configuration*** *page of the* ***/login*** *administrative interface.* ***It is highly recommended that HTTP remain disallowed as a security best practice****.*

Parameters for Client Scripting API

type=rep	The Bomgar client to which the command applies. Currently the API only supports **rep** as the client type.
operation=generate	The operation to perform. Currently the API only supports **generate** as the operation.
action=<command>&parameter=[value]	The name of the command to run and the necessary parameters. Available actions include: • login • start_jump_item_session Two actions are automatically added to the BRCS file: **login** and **delete_script_file**. The **delete_script_file** action has no parameters.

API Script Command: login

When generating any Bomgar Console Script, the **login** command is automatically added as the first command in the script file. It does not need to be specified in the URL used to generate the script file.

By default, this command opens the access console and attempts to log in using the credentials saved locally in the access console. If no credentials are saved, the command simply opens the access console login prompt. Once the user has correctly authenticated, the script continues running.

The **login** command has no effect if a user is already logged into the access console.

If you wish to specify the credentials to be used, you can create a separate script specifically to be used for logging in. The **login** command passes the login mechanism along with a username and password. Both username and password parameters are sent in plain text, unencrypted.

IMPORTANT!

*You cannot specify multiple commands in the URL used to generate a script. For example, you cannot specify **login** and multiple **start_jump_item_session** commands in the same URL. Each command must be generated as a separate script.*

*However, a skilled developer may edit the **.brcs** script file once it has been generated in order to modify the login credentials and then run another command. Bomgar does not support scripts modified in this manner.*

Optional Parameters for login

mechanism=[string]	The mechanism to use for authentication. Currently, only **username_password** is supported. If this parameter is supplied, both other parameters must also be supplied.
username=[string]	The username of the account with which to log in. If this parameter is supplied, both other parameters must also be supplied.
password=[string]	The password of the account with which to log in. If this parameter is supplied, both other parameters must also be supplied.

Query Examples: login

Log into the access console, specifying the username and password	https://access.example.com/api/client_script?type=rep&operation=generate&action=login&mechanism=username_password&username=username&password=password

API Script Command: start_jump_item_session

The **start_jump_item_session** command attempts to start a session with a Bomgar Jump Item. Users may run this command for all Jump Items they are permitted to access via the Jump management interface in the access console.

Optional Parameters for the start_jump_item_session Command

client.comments	If specified, only Jump Items with the given comments are included in the results. This field has a maximum length of 255 characters. Search is partial and case-insensitive.
client.hostname	If specified, only Jump Items with the given hostname are included in the results. This field has a maximum length of 255 characters. Search is partial and case-insensitive.
client.private_ip	If specified, only Jump Clients with the given private IP address are included in the results. This search field applies only to pinned clients. This field has a maximum length of 255 characters. Search is partial and case-insensitive.
client.public_ip	If specified, only Jump Clients with the given public IP address are included in the results. This search field applies only to pinned clients. This field has a maximum length of 255 characters. Search is partial and case-insensitive.
client.tag	If specified, only Jump Items with the given tag are included in the results. This field has a maximum length of 255 characters. Search is partial and case-insensitive.
jump.method	If specified, only Jump Items using the designated Jump method are included in the results. Acceptable values for this field are **push** (remote push), **local_push**, **pinned** (Jump Client), **rdp**, and **shelljump**.
session.custom.[custom field]=[string]	The code name and value of any custom fields. These fields must first be configured in **/login > Management > API Configuration**. Each attribute must be specified as a different parameter. Each custom field has a maximum length of 1024 characters. The maximum total size of all combined custom fields, including the external key, must be limited to 10KB.

IMPORTANT!

*At least one **client.*** parameter must be specified. If multiple **client.*** parameters are specified, then only clients matching all criteria are returned.*

Query Examples: start_jump_item_session

Start a session with a Jump Item whose hostname contains "ABCDEF02"	https://access.example.com/api/client_script?type=rep&operation=generate&action=start_jump_item_session&client.hostname=ABCDEF02
Start a session with a Jump Item whose comments contain "maintenance" and whose tag contains "server"	https://access.example.com/api/client_script?type=rep&operation=generate&action=start_jump_item_session&client.comments=maintenance&client.tag=server
Start a session with a pinned Jump Client whose private IP address begins with "10.10.24" and associate custom attributes with the session	https://access.example.com/api/client_script?type=rep&operation=generate&action=start_jump_item_session&client.private_ip=10.10.24&jump.method=pinned&session.custom.custom_field1=Custom%20Value&session.custom.custom_field2=123

***Note:** If more than one Jump Item matches the search criteria, then a dialog will open, giving the user the option to select the appropriate Jump Item.*

Reporting API

The Bomgar reporting API is designed to enable you to pull reporting data in XML format, suitable for importing into external databases and applications. The data presented is the same as in the session reports of the **/login** administrative interface.

XML data is pulled by sending a simple HTTP request to the Bomgar Appliance. The request can be sent using any HTTPS-capable socket library or scripting language module, a web browser, or a URL fetcher such as **cURL** or **wget**. Either **GET** or **POST** may be used as the request method.

IMPORTANT!

When making consecutive API calls, you must close the connection after each API call.

***Note:** By default, access to the API is SSL-encrypted; however, you can choose to allow HTTP access by checking the **Allow HTTP Access to XML API** option on the **Management > API Configuration** page of the **/login** administrative interface. **It is highly recommended that HTTP remain disallowed as a security best practice**.*

The reporting API URL is **https://access.example.com/api/reporting**.

An XML schema which formally describes the format of the returned reporting data is available at **https://access.example.com/api/reporting.xsd**.

In order to issue an API request to the reporting API, you will need to supply the username and password for a Bomgar user account. That account must have the reporting permission **Allowed to use reporting API**. The account must also have one or more of the following permissions, depending upon which type of reports you wish to run: **Allowed to View Access Session Reports** for only their sessions, their teams' sessions, or all sessions; and **Allowed to view access session recordings**.

Required Parameters for Reporting API

username=[string]	The username to use when retrieving the reports. This user must have permission to use the reporting API. The user must also have permission to view reports. Reports returned will depend on the user's specific reporting permissions.
password=[string]	The password associated with this username.
generate_report=[string]	The type of report to be generated. Report types can be any of the following: AccessSession AccessSessionListing AccessSessionRecording AccessSessionSummary CommandShellRecording Team

The reporting API returns XML responses that declare a namespace. If you are parsing these responses with a namespace-aware parser, you will need to set the namespace appropriately or ignore the namespace while parsing the XML.

- Reporting API: https://www.bomgar.com/namespaces/API/reporting

***Note:** The above namespace is returned XML data and is not a functional URL.*

Download Reports with AccessSession

The **AccessSession** query returns full information for all sessions which match given search parameters. You may use any of the following sets of parameters to generate reports:

- **start_date** and **duration**
- **start_time** and **duration**
- **end_date** and **duration**
- **end_time** and **duration**
- **lsid**
- **lsids**

Parameters for AccessSession

start_date=[YYYY-MM-DD]	Specifies that the report should return all sessions, even those still in progress, that began on or after this date and that are within the duration specified below.
start_time=[timestamp]	Specifies that the report should return all sessions, even those still in progress, that began at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
end_date=[YYYY-MM-DD]	Specifies that the report should return only closed sessions that ended on or after this date and that are within the duration specified below.
end_time=[timestamp]	Specifies that the report should return only closed sessions that ended at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
duration=[integer]	Length of time from the specified date or time for which you wish to pull reports, or **0** to pull from the specified date to present. If **start_date** or **end_date** is specified, **duration** will represent days; if **start_time** or **end_time** is specified, **duration** will represent seconds.
lsid=[string]	The ID of the session for which you wish to see details.
lsids=[comma-separated strings]	A comma-delimited list of the IDs of sessions for which you wish to see details.

XML Response for AccessSession Query

<session_list>	Contains a **<session>** element for each session that matches the given criteria. If no sessions are returned, this element will contain no **<session>** elements. If an error occurs during the search, it will contain an **<error>** element describing the problem.

Element Names and Attributes

/session_list/session

lsid (attribute)	A string which uniquely identifies this session.

<session_type>	Indicates the type of session for which the report was run. The value will always be **support** in the current Bomgar API version.
<lseq>	An incrementing number used to represent sessions in a non-string format. ***Note:*** *The LSEQ element is not guaranteed to be unique or strictly sequential.*
<start_time>	The date and time the session was begun. Data is returned in ISO 8601 format. Also contains a **timestamp** attribute which displays the start time as a UNIX timestamp (UTC).
<end_time>	The date and time the session was ended. Data is returned in ISO 8601 format. Also contains a **timestamp** attribute which displays the end time in UNIX timestamp (UTC). This element will be empty for sessions which are still in progress when the report was run or which closed abnormally.
<duration>	Session length in HH:MM:SS format.
<jumpoint>	The name of the Jumpoint through which this session was initiated, if any. Also contains an **id** attribute, which displays the unique ID assigned to the Jumpoint.
<custom_attributes>	Contains a **<custom_attribute>** element for each custom field assigned to a session. This element displays only if custom fields have been defined. The format of each **<custom_attribute>** element is described below.
<session_chat_view_url>	The URL at which this session's chat transcript can be viewed in a web browser. This element is displayed only for sessions that have successfully ended.
<session_chat_download_url>	The URL at which this session's chat transcript can be downloaded. This element is displayed only for sessions that have successfully ended.
<session_recording_view_url>	The URL at which the video of the session may be viewed in a web browser. This element is displayed only if screen sharing recording was enabled at the time of the session and only if the user initiated screen sharing during the session. It is available only for sessions that have successfully ended.
<session_recording_download_url>	The URL at which the video of the session may be downloaded. This element is displayed only if screen sharing recording was enabled at the time of the session and only if the user initiated screen sharing during the session. It is available only for sessions that have successfully ended.
<command_shell_recordings>	Contains a **<command_shell_recording>** element for each command shell that was initiated during the session. This element is displayed only if the user opened a remote command shell during the session, if command shell recording was enabled at the time of the session, and if the requesting user has permission to view session recordings. Each **<command_shell_recording>** element contains the child elements **<download_url>** and **<view_url>** as described below.
<file_transfer_count>	The number of file transfers which occurred during the session.
<file_move_count>	The number of files renamed via the **File Transfer** interface during the session.
<file_delete_count>	The number of files deleted via the **File Transfer** interface during the session.

<primary_customer>	Lists the **gsnumber** as an attribute and as an element, the **name** of the remote endpoint accessed by the user.
<primary_rep>	Lists the **gsnumber** and **id** as attributes and as an element, the **name** of the user who owned the session.
<customer_list>	A list of all endpoints accessed in the session. There should always be exactly one endpoint per session in the current Bomgar API version. The format of each **<customer>** element is described below.
<rep_list>	A list of all users who participated in the session, whether as the session owner or as conference members. The format of each **<representative>** element is described below.
<session_details>	Contains a chronological list of all events which occurred during the session. This element contains one or more child **<event>** elements, described below.

/session_list/session/custom_attributes/custom_attribute

display_name (attribute)	The display name assigned to the custom attribute.
code_name (attribute)	The code name assigned to the custom attribute.

/session_list/session/command_shell_recordings/command_shell_recording

instance (attribute)	The instance of the command shell session, starting with **0**.
<download_url>	The URL at which the video of the command shell session may be downloaded.
<view_url>	The URL at which the video of the command shell session may be viewed in a web browser.

/session_list/session/customer_list/customer

gsnumber (attribute)	Uniquely identifies the endpoint in regards to its current connection to the Bomgar Appliance. A gsnumber may be recycled, so while two endpoints connected at the same time will never have the same gsnumber, one endpoint may have a gsnumber that was assigned to another endpoint in the past. Can be used to correlate a **<customer>** element with a **<primary_customer>** or with an event's **<performed_by>** or **<destination>** element.
<username>	The name used to identify the endpoint during the session.
<public_ip>	The endpoint's public IP address.
<private_ip>	The endpoint's private IP address.
<hostname>	The hostname of the endpoint.
<os>	The operating system of the endpoint.

/session_list/session/rep_list/representative

gsnumber (attribute)	Uniquely identifies the user in regards to their current connection to the Bomgar Appliance. A gsnumber is assigned on a per-connection basis, so if a user leaves a session and then rejoins without logging out of the Bomgar Appliance, their gsnumber will remain the same. However, if the user's connection is terminated for any reason, when that user logs back into the Bomgar Appliance, they will be assigned a new gsnumber and will also appear multiple times in the <**rep_list**> element. A gsnumber may be recycled, so while two people connected at the same time will never have the same gsnumber, one person may have a gsnumber that was assigned to another person in the past. Can be used to correlate a <**representative**> element with a <**primary_rep**> or with an event's <**performed_by**> or <**destination**> element.
id (attribute)	Unique ID assigned to the user.
<username>	The username assigned to the user.
<display_name>	The display name assigned to the user. Note that this field contains the display name's value at the time of the conference, which may not match the current value if the **display_name** has subsequently been changed.
<public_ip>	The user's public IP address.
<private_ip>	The user's private IP address.
<hostname>	The hostname of the user's computer.
<os>	The operating system of the user's computer.
<session_owner>	Integer value (**1** or **0**) indicating whether the user was the owner of the session or was merely a conference member.
<seconds_involved>	Integer value indicating the number of seconds the user was involved in this session.
<invited>	Integer value (**1**) present only if the user is an invited user.

/session_list/session/session_details/event

timestamp (attribute)	The system time at which the event occurred.

<table>
<tr><td>event_type (attribute)</td><td>The type of event which occurred. Event types include the following:
<table>
<tr><td>Chat Message</td><td>Registry Imported</td></tr>
<tr><td>Command Shell Session Started*</td><td>Registry Key Added</td></tr>
<tr><td>Conference Member Added</td><td>Registry Key Deleted</td></tr>
<tr><td>Conference Member Departed</td><td>Registry Key Renamed</td></tr>
<tr><td>Conference Member State Changed</td><td>Registry Value Added</td></tr>
<tr><td>Conference Owner Changed</td><td>Registry Value Deleted</td></tr>
<tr><td>Credential Injection Attempt</td><td>Registry Value Modified</td></tr>
<tr><td>Credential Injection Attempt Failed</td><td>Registry Value Renamed</td></tr>
<tr><td>Directory Created</td><td>Screen Recording</td></tr>
<tr><td>File Deleted</td><td>Screenshot Captured</td></tr>
<tr><td>File Download</td><td>Service Access Allowed</td></tr>
<tr><td>File Download Failed</td><td>Session End</td></tr>
<tr><td>File Moved</td><td>Session Foreground Window Changed</td></tr>
<tr><td>File Upload</td><td>Session Start</td></tr>
<tr><td>File Upload Failed</td><td>System Information Retrieved</td></tr>
<tr><td>Registry Exported</td><td></td></tr>
</table>
*Will only appear if recording is enabled for this session.</td></tr>
<tr><td><performed_by></td><td>The entity that performed the action. Indicates the entity's gsnumber and also its type, indicating whether this action was performed by the system, a endpoint, or a representative.</td></tr>
<tr><td><destination></td><td>The entity to which the event was directed. Indicates the entity's gsnumber and also its type, indicating whether this action was directed to the system, a customer, or a user.</td></tr>
<tr><td><body></td><td>The text of the message as displayed in the chat log area.</td></tr>
<tr><td><encoded_body></td><td>Can be shown in place of the <body> element above. Contains the base64 (RFC 2045 section 6.8) encoded value of what would have been shown in the <body> element, and is shown ONLY if the <body> text contains characters that are invalid according to XML specification. These characters are typically the result of binary data being sent through chat messages.</td></tr>
<tr><td><filename></td><td>The name of the transferred file.</td></tr>
<tr><td><files></td><td>If this event involved the transferring of files, then this element will contain a <file> element for every file transferred.</td></tr>
<tr><td><filesize></td><td>An integer indicating the size of the transferred file.</td></tr>
</table>

<system_information>	Applies only to **System Information Retrieved** events wherein the system information is pulled automatically upon session start. This element contains multiple **<category>** child elements as described below. ***Note:*** *System information is logged only when pulled automatically at the beginning of the session and not when specifically requested by the user. This is to prevent overload with the large amount of dynamic data that can be retrieved from the remote system.*
<data>	Contains an arbitrary number of **<value name="_" value="_" />** elements. The name and number of these elements varies based on **event_type**. For example, when a user joins the session, a **Conference Member Added** event would contain **<value>** elements for the user's **name**, **private_ip**, **public_ip**, **hostname**, and **os**.

/session_list/session/session_details/event/system_information/category

<description>	Contains multiple **<field>** elements, each of which contains a descriptor for the specific data field. For example, the **Drives** category would have **<field>** elements **Drive**, **Type**, **Percent Used**, etc. These **<field>** elements can be compared to table header cells.
<data>	Contains multiple **<row>** elements, each of which contains multiple **<field>** elements that correspond to the **<field>** elements above. For example, the **Drives** category would have a separate **<row>** for each drive on the endpoint computer. An example **<row>** might contain **<field>** elements **C:**, **Local Disk**, **60%**, etc. These **<row>** elements can be compared to table rows, with each **<field>** element a table cell.

Query Examples for AccessSession

Sessions started March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSession&start_date=2015-03-01&duration=0
Sessions started the month of March 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSession&start_date=2015-03-01&duration=31
Sessions started 8:00 AM March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSession&start_time=1425196800&duration=0
Sessions started 8:00 AM March 1 2015 to 6:00 PM March 1 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSession&start_time=1425196800&duration=36000
Sessions ended March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSession&end_date=2015-03-01&duration=0

Sessions ended the month of March 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSession&end_date=2015-03-01&duration=31
Sessions ended 8:00 AM March 1 2015 to 6:00 PM March 1 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSession&end_time=1425196800&duration=36000
Session c69a8e10bea9428f816cfababe9815fe	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSession&lsid=c69a8e10bea9428f816cfababe9815fe
Sessions c69a8e10bea9428f816cfababe9815fe, a5eeaa58591047b88556f944804227b0, 5bf07601298b495b87310da9ce571e22	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSession&lsids=c69a8e10bea9428f816cfababe9815fe,a5eeaa58591047b88556f944804227b0,5bf07601298b495b87310da9ce571e22

Download Reports with AccessSessionListing

The **AccessSessionListing** query returns a list of session IDs, external keys, and availability of a recording for sessions which match given search parameters. You may use any of the following sets of parameters to generate reports:

- **start_date** and **duration**
- **start_time** and **duration**
- **end_date** and **duration**
- **end_time** and **duration**

Parameters for AccessSessionListing

start_date=[YYYY-MM-DD]	Specifies that the report should return all sessions, even those still in progress, that began on or after this date and that are within the duration specified below.
start_time=[timestamp]	Specifies that the report should return all sessions, even those still in progress, that began at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
end_date=[YYYY-MM-DD]	Specifies that the report should return only closed sessions that ended on or after this date and that are within the duration specified below.
end_time=[timestamp]	Specifies that the report should return only closed sessions that ended at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
duration=[integer]	Length of time from the specified date or time for which you wish to pull reports, or **0** to pull from the specified date to present. If **start_date** or **end_date** is specified, **duration** represents days; if **start_time** or **end_time** is specified, **duration** represents seconds.

XML Response for AccessSessionListing Query

<session_summary_list>	Contains a **<session_summary>** element for each session that matches the given criteria. If no sessions are returned, this element will contain no **<session_summary>** elements. If an error occurs during the search, it will contain an **<error>** element describing the problem.

Element Names and Attributes

/session_summary_list/session_summary

lsid (attribute)	The session ID for the given session.
has_recording (attribute)	Integer (**1** or **0**) indicating if the given session has a session recording.
external_key (attribute)	An arbitrary string that can link this session to an identifier on an external system, such as a customer relationship management ticket ID. This can be input from within the access console or defined programmatically. This element is displayed only if an external key has been defined.

Query Examples for AccessSessionListing

Sessions started March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionListing&start_date=2015-03-01&duration=0
Sessions started the month of March 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionListing&start_date=2015-03-01&duration=31
Sessions started 8:00 AM March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionListing&start_time=1425196800&duration=0
Sessions started 8:00 AM March 1 2015 to 6:00 PM March 1 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionListing&start_time=1425196800&duration=36000
Sessions ended March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionListing&end_date=2015-03-01&duration=0
Sessions ended the month of March 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionListing&end_date=2015-03-01&duration=31
Sessions ended 8:00 AM March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionListing&end_time=1425196800&duration=0
Sessions ended 8:00 AM March 1 2015 to 6:00 PM March 1 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionListing&end_time=1425196800&duration=36000

Download Reports with AccessSessionSummary

The **AccessSessionSummary** query returns an overview of access session statistics by user. You may use any of the following sets of parameters to generate reports:

- **start_date**, **duration**, and **report_type**
- **start_time**, **duration**, and **report_type**
- **end_date**, **duration**, and **report_type**
- **end_time**, **duration**, and **report_type**

Parameters for AccessSessionSummary

start_date=[YYYY-MM-DD]	Specifies that the report should return all sessions, even those still in progress, that began on or after this date and that are within the duration specified below.
start_time=[timestamp]	Specifies that the report should return all sessions, even those still in progress, that began at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
end_date=[YYYY-MM-DD]	Specifies that the report should return only closed sessions that ended on or after this date and that are within the duration specified below.
end_time=[timestamp]	Specifies that the report should return only closed sessions that ended at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
duration=[integer]	Length of time from the specified date or time for which you wish to pull reports, or 0 to pull from the specified date to present. If **start_date** or **end_date** is specified, **duration** represents days; if **start_time** or **end_time** is specified, **duration** represents seconds.
report_type=[string]	In the current Bomgar API version, **user** is the only accepted value.

XML Response for AccessSessionSummary Query

<summary_list>	Contains a **<summary>** element for each record that matches the given criteria. If no sessions are returned, this element will contain no **<summary>** elements. If an error occurs during the search, it will contain an **<error>** element describing the problem.

Element Names and Attributes

/summary_list/summary

id (attribute)	Returns the user's unique ID.
type (attribute)	Specifies the report type generated. This value is always **user** in the current API version.

<display_name>	The display name of the user. Note that since summary reports represent an aggregation of sessions over a period of time, the display name used is the current value for the user, which may have been edited since the time of the first returned session.
<total_sessions>	The total number of sessions run by the user in the time specified.
<avg_sessions_per_weekday>	The average number of sessions conducted on Monday through Friday by the user, expressed as a decimal rounded to the nearest point.
<avg_duration>	The average length of each session, expressed as HH:MM:SS.

Query Examples

Sessions started March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionSummary&start_date=2015-03-01&duration=0&report_type=user
Sessions started the month of March 2015, by user	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionSummary&start_date=2015-03-01&duration=31&report_type=user
Sessions started 8:00 AM March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionSummary&start_time=1425196800&duration=0&report_type=user
Sessions started 8:00 AM March 1 2015 to 6:00 PM March 1 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionSummary&start_time=1425196800&duration=36000&report_type=user
Sessions ended March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionSummary&end_date=2015-03-01&duration=0&report_type=user
Sessions ended the month of March 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionSummary&end_date=2015-03-01&duration=31&report_type=user
Sessions ended 8:00 AM March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionSummary&end_time=1425196800&duration=0&report_type=user
Sessions ended 8:00 AM March 1 2015 to 6:00 PM March 1 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionSummary&end_time=1425196800&duration=36000&report_type=user

Download Reports with AccessSessionRecording

The **AccessSessionRecording** query returns the requested access session recording file. Depending on your browser, this query will either immediately begin download or prompt you to open or save the file. Note that the requesting user must have permission to view session recordings.

Parameter for AccessSessionRecording

lsid=[string]	The session ID for which you wish to download the video recording of the session.

Query Example for AccessSessionRecording

AccessSessionRecording: Session c69a8e10bea9428f816cfababe9815fe	https://access.example.com/api/reporting?username=test&password=test&generate_report=AccessSessionRecording&lsid=c69a8e10bea9428f816cfababe9815fe

Download Reports with CommandShellRecording

The **CommandShellRecording** query returns the requested command shell recording. Depending on your browser, this query will either immediately begin download or prompt you to open or save the file. Note that the requesting user must have permission to view session recordings.

Parameters for CommandShellRecording

lsid=[string]	The session ID for which you wish to download the video recording of the command shell.
instance=[integer]	The instance number of the command shell recording you wish to download. Instances are enumerated starting with **0**. The instance number can be obtained from the **AccessSession** report.

Optional Parameter for CommandShellRecording

format=[string]	If this parameter has the value of **txt**, the command shell output will be in a text format instead of a recording.

Query Examples for CommandShellRecording

CommandShellRecording: First shell instance of session c69a8e10bea9428f816cfababe9815fe	https://access.example.com/api/reporting?username=test&password=test &generate_report=CommandShellRecording&lsid= c69a8e10bea9428f816cfababe9815fe&instance=0
CommandShellRecording: Third shell instance of session c69a8e10bea9428f816cfababe9815fe	https://access.example.com/api/reporting?username=test&password=test &generate_report=CommandShellRecording&lsid= c69a8e10bea9428f816cfababe9815fe&instance=2

Download Reports with Team

The **Team** query returns information about activity within a team. You may use any of the following sets of parameters to generate reports:

- **start_date** and **duration**
- **start_time** and **duration**
- **end_date** and **duration**
- **end_time** and **duration**

Parameters for Team

start_date=[YYYY-MM-DD]	Specifies that the report should return team activity that began on or after this date and that is within the duration specified below.
start_time=[timestamp]	Specifies that the report should return team activity that began at or after this time and that is within the duration specified below. The time must be a UNIX timestamp (UTC).
end_date=[YYYY-MM-DD]	Specifies that the report should return team activity that ended on or after this date and that is within the duration specified below.
end_time=[timestamp]	Specifies that the report should return team activity that ended at or after this time and that is within the duration specified below. The time must be a UNIX timestamp (UTC).
duration=[integer]	Length of time from the specified date or time for which you wish to pull reports, or **0** to pull from the specified date to present. If **start_date** or **end_date** is specified, **duration** will represent days; if **start_time** or **end_time** is specified, **duration** will represent seconds.

Optional Parameter for Team

team_id=[integer]	The numeric ID of the team by which to filter results. Only the activity within the specified team will be returned. If this parameter is not specified, results from all teams will be returned.

XML Response for Team Query

<team_activity_list>	Contains a **<team_activity>** element for each team with any activity within the given parameters. If no teams are returned, this element will contain no **<team_activity>** elements. If an error occurs during the search, it will contain an **<error>** element describing the problem. Also contains **<start_time>** and **<end_time>** elements displaying the time parameters in the system time and with a **timestamp** attribute in UTC.

Element Names and Attributes

/team_activity_list/team_activity

id (attribute)	Integer representing the team's unique ID.
name (attribute)	The display name of the team. Note that this field contains the team name as it currently appears, which may not match the value at the time of the conference if the team name has been subsequently changed.
<logged_in_privileged users>	Contains a **<representative>** element for each user in that team who was logged into the access console before the first event in the report occurred. If no users were logged in at the start time, this element will be empty.
<events>	Contains an **<event>** element for each event that occurred within this team.

/team_activity_list/team_activity/logged_in_representatives/representative

gsnumber (attribute)	Uniquely identifies the user in regards to their current connection to the Bomgar Appliance. A gsnumber is assigned on a per-connection basis, so if a user leaves a session and then rejoins without logging out of the Bomgar Appliance, their gsnumber will remain the same. However, if the user's connection is terminated for any reason, when that user logs back into the Bomgar Appliance, they will be assigned a new gsnumber. A gsnumber may be recycled, so while two people connected at the same time will never have the same gsnumber, one person may have a gsnumber that was assigned to another person in the past. Can be used to correlate a **<representative>** element with an event's **<performed_by>** or **<destination>** element.
id (attribute)	Unique ID assigned to the user.
<display_name>	The display name assigned to the user. Note that this field contains the display name's value at the time of the conference, which may not match the current value if the **display_name** has subsequently been changed.
<public_ip>	The user's public IP address.
<private_ip>	The user's private IP address.

/team_activity_list/team_activity/events/event

timestamp (attribute)	The system time at which the event occurred.
event_type (attribute)	The type of event which occurred. Event types include the following: Chat Message Conference Member Added Conference Member Departed Conference Member State Changed File Download File Download Failed File Upload File Upload Failed Files Shared Jump Item Authorization Request Jump Item Authorization Request Utilized Pinned Session Moved Away from Queue Pinned Session Moved to Queue Representative Monitoring Started Representative Monitoring Stopped Session Deployed to Queue Session Undeployed from Queue
<performed_by>	The entity that performed the action. Indicates the entity's **gsnumber** and also its **type**, indicating whether this entity was the system or a user.
<destinations>	If this event was targeted to one or more specific users, it will contain one or more **<destination>** elements as described below.
<files>	If this event involved the transfer of files, then this element will contain a **<file>** element for every file transferred.
<data>	Contains an arbitrary number of **<value name="_" value=" _" />** elements. The name and number of these elements varies based on the **event_type**. For example, when a user logs into the access console, a **Conference Member State Changed** event would contain **<value>** elements for the **hostname**, **os**, **private_ip**, **public_ip**, and **state**.
<body>	The text of the chat message as displayed in the chat log area.
<encoded_body>	Can be shown in place of the **<body>** element above. Contains the base64 (RFC 2045 section 6.8) encoded value of what would have been shown in the **<body>** element, and is shown ONLY if the **<body>** text contains characters that are invalid according to XML specification. These characters are typically the result of binary data being sent through chat messages.

/team_activity_list/team_activity/events/event/destinations/destination

gsnumber (attribute)	Indicates the **gsnumber** of the entity to which the event was destined.
type (attribute)	Indicates whether this entity was the system or a user.
[value]	The name of the entity to which the event was destined.

/team_activity_list/team_activity/events/event/files/file

name (attribute)	The name of the transferred file.
size (attribute)	An integer indicating the size of the transferred file.

Query Examples for Team

Activity started March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=Team&start_date=2015-03-01&duration=0
Activity started the month of March 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=Team&start_date=2015-03-01&duration=31
Activity started 8:00 AM March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=Team&start_time=1425196800&duration=0
Activity started 8:00 AM March 1 2015 to 6:00 PM March 1 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=Team&start_time=1425196800&duration=36000
Activity started March 1 2015 to present for a specific team	https://access.example.com/api/reporting?username=test&password=test&generate_report=Team&start_date=2015-03-01&duration=0&team_id=1
Activity ended March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=Team&end_date=2015-03-01&duration=0
Activity ended the month of March 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=Team&end_date=2015-03-01&duration=31
Activity ended 8:00 AM March 1 2015 to present	https://access.example.com/api/reporting?username=test&password=test&generate_report=Team&end_time=1425196800&duration=0
Activity ended 8:00 AM March 1 2015 to 6:00 PM March 1 2015	https://access.example.com/api/reporting?username=test&password=test&generate_report=Team&end_time=1425196800&duration=36000
Activity ended March 1 2015 to present for a specific team	https://access.example.com/api/reporting?username=test&password=test&generate_report=Team&end_date=2015-03-01&duration=0&team_id=1

Backup API

The Bomgar backup API is designed to enable you to automatically back up your Bomgar software configuration on a recurring basis. The backup file will include all your configuration settings and logged data except for recordings and some large files from the file store. The backup will only include files from the file store less than 200 KB in size and no more than 50 files total. In the event of a hardware failure, having a backup file will help to speed the disaster recovery process.

Commands are executed by sending a simple HTTP request to the Bomgar Appliance. The request can be sent using any HTTPS-capable socket library or scripting language module, a web browser, or a URL fetcher such as **cURL** or **wget**. Either **GET** or **POST** may be used as the request method.

Note: *By default, access to the API is SSL-encrypted; however, you can choose to allow HTTP access by checking the* ***Allow HTTP Access to XML API*** *option on the* ***Management > API Configuration*** *page of the* ***/login*** *administrative interface.* ***It is highly recommended that HTTP remain disallowed as a security best practice****.*

The backup API URL is **https://access.example.com/api/backup**.

Required Parameters for Backup API

username=[string]	The username to use when backing up the site. Must be an administrator.
password=[string]	The password associated with this username.

Query Example

backup	https://access.example.com/api/backup?username=test&password=test

Test Scenario

To get started with this basic API integration, follow the steps below.

1. Log into your Bomgar administrative interface and go to **Management > API Configuration**. Check the box to **Enable XML API**. If you do not have a valid SSL certificate, you may need to enable the option to **Allow HTTP Access to XML API** while you are testing.
2. Create a special Bomgar user account to be used for API commands. Give this user a password that does not need to be reset and never expires, and enable all necessary permissions such as the ability to view reports, generate session keys, and participate in the general queue. For simplicity, you can make this user an administrator with all permissions enabled, though this is not required.
3. Create a normal user account for yourself. Download the Bomgar access console and log in.
4. You now can begin testing API commands using your browser. Create the appropriate URLs by copying the samples into a text editor. Modify the parameters as needed for your environment, replacing the hostname, username, password, external key, and so forth.
5. Paste the customized URLs into your browser to test the API commands. The appropriate XML should be returned in the browser.
6. If you receive any errors such as **Document Not Found**, check that the API user has the necessary permissions. Also, make sure that a user is logged into the site while you are testing.

API Change Log

API Version 1.14.0 for PAM 15.3.x

- Import Jump Item shortcuts to minimize the time needed to create Jump Items.
  - API Command: import_jump_shortcut

Privileged Access Management API Version Reference

The following table shows the relationship between the API and Bomgar versions for Privileged Access Management.

API Version	Bomgar Version
1.13.0	15.1.x, 15.2.x
1.14.0	15.3.x

Disclaimers, Licensing Restrictions and Tech Support

Disclaimers

This document is provided for information purposes only. Bomgar Corporation may change the contents hereof without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. Bomgar Corporation specifically disclaims any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. The technologies, functionality, services, and processes described herein are subject to change without notice.

BOMGAR, BOMGAR BOX, mark B, JUMP and UNIFIED REMOTE SUPPORT are trademarks of Bomgar Corporation; other trademarks shown are the property of their respective owners.

Licensing Restrictions

One Bomgar Privileged Access Management license enables one support representative at a time to troubleshoot an unlimited number of remote computers, whether attended or unattended. Although multiple accounts may exist on the same license, two or more licenses (one per concurrent support representative) are required to enable multiple support representatives to troubleshoot simultaneously.

One Bomgar Privileged Access Management license enables access to one endpoint system. Although this license may be transferred from one system to another if access is no longer required to the first system, two or more licenses (one per endpoint) are required to enable access to multiple endpoints simultaneously.

Tech Support

At Bomgar, we are committed to offering the highest quality service by ensuring that our customers have everything they need to operate with maximum productivity. Should you need any assistance, please contact Bomgar Technical Support at help.bomgar.com.

Technical support is provided with annual purchase of our maintenance plan.