SonicWall® SonicOS API 6.5.4

Reference

SONICWALL®

Contents

1

About SonicOS API

SonicOS API Function

SonicOS API provides an alternative to the SonicOS Command Line Interface (CLI) for configuring selected functions.

SonicOS API is disabled by default in SonicOS. Any attempt to access SonicOS API while it is disabled results in an `HTTP 403 Forbidden` error. To use the SonicOS API, you must enable it, either through the SonicOS Management Interface or from the CLI.

SonicOS API is supported on all platforms running SonicOS 6.5.4 and higher.

Topics:

Enabling through the Management Interface

To enable SonicOS API through the management interface:

1. Navigate to **MANAGE | System Setup | Appliance > Base Settings**.
2. Scroll down to the **SonicOS API** section.
3. Select **Enable SonicOS API**.
4. Click **Accept**.

Enabling through the CLI

Starting at the `config#` prompt:

```
config(<serial number>)# administration
(config-administration)# sonicos-api
(config-administration)# commit
```

Supported Request Methods

SonicOS API utilizes four of the methods defined in the HTTP protocol (RFC 7231 and RFC 5789) to create, read, update and delete (CRUD) resources. Supported HTTP request methods describes the supported HTTP methods for management operations after authentication. Refer to Client Authentication on page 12 for the methods supported during authentication.

Supported HTTP request methods

HTTP method	Description
GET	Retrieves the specified resource or collection of resources. GET is a read-only operation that does not alter appliance state or configuration. A GET operation should not contain a request-body.
POST	Submits data to be processed by the specified resource or collection of resources. In most cases, the POST verb is used by SonicOS APIs to create and add a resource to a collection of resources (for example, add a new MAC address-object to collection of objects).
PUT	Updates the specified resource. The data included in the PUT request-body replaces the previous configuration.
DELETE	Deletes the specified resource or collection of resources.

Supported HTTP header request and response formats

Type	Example
Text/plain	`GET /api/sonicos/address-objects/mac` `Accept: text/plain`

Application/JSON	`POST /api/sonicos/address-objects/mac`
	`Content-type: application/json`
	`Accept: application/json`
	`{`
	`"address_object": {`
	`"mac": {`
	`"name": "001122334455"`
	`,"address": "001122334455"`
	`,"multi_homed": true`
	`,"zone": "LAN"`
	`}`
	`}`
	`}`

To configure all other parameters:

```
config(C0EAE483FB86)# administration
(config-administration)# sonicos-api
(config-sonicos-api)# exit
(config-administration)# commit
```

SonicOS API Commands	
basic	chap
digest	hold-password
integrity-protection	max-nonce
md5-digest	public-key
rsa-key-size	rsa-padding-type
session-security	sha256-digest
two-factor-bearer-token	

Supported HTTP Headers

Content-type	Specifies the format (MIME type) of the request body (input).
Accept	Specifies the format of the response body (output).

Supported HTTP MIME Types

SonicOS supports these HTTP MIME types:

- Text/plain
- Application/JSON

These HTTP headers define the request and response format:

- **Content-type** – Specifies the format (MIME type) of the request body (input)
- **Accept** – Specifies the format of the response body (output)

NOTE: The headers can be used to obtain mixed input/output. See examples below for reference.

Examples

Topics:

Application/JSON

When specified, the request and/or response body is expected to be in SonicOS API JSON format.

Request

```
POST /api/sonicos/address-objects/mac
Content-type: application/json
Accept: application/json

{
```

```
     "address_object": {
         "mac": {
              "name": "001122334455"
             ,"address": "001122334455"
             ,"multi_homed": true
             ,"zone": "LAN"
         }
     }
}
```

Response

```
HTTP/1.0 200 OK
Server: SonicWALL
Content-type: application/json; charset=UTF-8

{
     "status": {
        "success": true
        ,"cli": {
             "depth": 1
            ,"mode": "config_mode"
            ,"configuring": true
            ,"pending_config": true
            ,"restart_required": "NONE"
        }
        ,"info": [
             { "level": "info", "code": "E_OK", "message": "Success." }
        ]
    }
}
```

Text/Plain

When specified, the request and/or response body is expected to be in SonicOS CLI plain-text command format.

Topics:

Request 1

```
GET /api/sonicos/address-objects/mac
Accept: text/plain
```

Response

```
HTTP/1.0 200 OK
Server: SonicWALL
Content-type: text/plain; charset=UTF-8

address-object mac example address 001122334455
    zone LAN
    multi-homed
    exit
```

Request 2

```
POST /api/sonicos/direct/cli
Content-type: text/plain
Accept: application/json
```

```
address-object mac example address 001122334455
   zone LAN
   multi-homed
   exit
```

Response

```
HTTP/1.0 200 OK
Server: SonicWALL
Content-type: application/json; charset=UTF-8

{
     "status": {
        "success": true
        ,"cli": {
             "depth": 1
            ,"mode": "config_mode"
            ,"configuring": true
            ,"pending_config": true
            ,"restart_required": "NONE"
        }
        ,"info": [
             { "level": "info", "code": "E_OK", "message": "Success." }
        ]
    }
}
```

Status and Error Representation

All plain text output from the last backend CLI command executed is captured and returned to the client. If the command executed was not a show command and the requested operation succeeded, then the response body is empty. This is consistent with the CLI when executing a command via SSH or the serial console in that status is only rendered to the console upon error.

A JSON status object is guaranteed to be returned in the response body when performing a POST, PUT, or DELETE operation or upon error(s) encountered when processing a request.

Topics:

HTTP Status Codes

SonicOS API uses standard HTTP status codes to report success or failure when servicing a request.

HTTP Status Codes

Code	Status Text	Description
200	OK	The request succeeded.
400	Bad Request	An invalid request was submitted. Verify that the request URI is correct and that the request body is as expected.
401	Not Authorized	The user is unauthenticated or lacks the required privileges for the requested operation. This may be accompanied by headers for initiating authentication, depending on the scheme(s) enabled for that.
403	Forbidden	The request was understood by the server but denied. The response body notes the reason why the request was denied.

HTTP Status Codes

Code	Status Text	Description
404	Not Found	The resource specified was not found.
405	Method Not Allowed	The HTTP verb specified is not allowed or supported by the resource specified.
406	Not Acceptable	The MIME type specified in the HTTP `Content-type` and/or `Accept` header is not supported.
413	Request body too large	Maximum size of the request body was exceeded.
414	Request URL too long	The request URL exceeded the maximum size allowed or contains extra/unknown parameters (directories).
500	Internal Server Error	The request failed due to an internal server error. The response body should note the reason why the request failed.
503	No resources	Maximum number of sessions was exceeded.

Application/JSON

A JSON status object is guaranteed to be returned in the response body when performing a POST, PUT, or DELETE operation or upon error(s) encountered when processing a request.

Topics:

Schema Structure

```
{
   "status": {
      "success": {boolean}
      ,"cli": {
         "depth": {number}
         ,"mode": "{string}"
         ,"command": "{string}"
         ,"configuring": {boolean}
         ,"pending_config": {boolean}
         ,"restart_required": "{string}"
      }
      ,"info": [
         { "level": "{string}", "code": "{string}", "message": "{string}" }
         ...
      ]
   }
}
```

Schema Attributes

Schema attributes

Attribute	Type	Description
`status`	object	Status object.
`status.success`	boolean (true|false)	Boolean success flag. Refer to the `status.info` array for more detailed information as to what caused the error if the success flag is false.
`status.cli`	object	CLI status. **NOTE:** This attribute is included only when an API sent one or more commands to the CLI backend.
`status.cli.depth`	number (uint8)	Current mode depth of the CLI: • 0 = top-level mode • >= 1 config mode
`status.cli.mode`	string	Name of the current mode.
`status.cli.command`	string	Command last executed. **NOTE:** This attribute is only included upon command error(s).
`status.cli.configuring`	boolean (true|false)	Boolean configuring flag. Should always be true upon one or more consecutive POST, PUT or DELETE API calls that modify the configuration.
`status.cli.pending_config`	boolean (true|false)	Boolean pending-config flag. Should always be true upon one or more consecutive POST, PUT or DELETE API calls that modify the configuration. This flag should be cleared once any/all pending changes are committed (saved).
`status.cli.restart_required`	string	Appliance restart status. To take effect, some configuration changes require an appliance restart. These values indicate the type of restart needed: • NONE • APPLIANCE • CHASSIS • CHASSIS_SHUTDOWN • ALL_BLADES
`status.info`	array	Informational message(s).
`status.info.level`	string	Status level: info, warning, error.
`status.info.code`	string	Status code. If success, E_OK is returned, else E_{*XXX*} where *XXX* = error code.
`status.info.message`	string	Status message.

Client Authentication

SonicOS API currently offers the following mechanisms for initial client authentication:

- HTTP Basic Authentication (RFC 2617)
- HTTP Digest Access Authentication (RFC-7616)
- Public Key Authentication
- Challenge-Handshake Authentication (CHAP)

- Time-Based One-Time Password (TOTP)/Bearer Token Authentication

Regardless of the authentication mechanism used, only:

- A single administrator can manage (modify configuration) at any given time. This remains true regardless of where an admin logged in (web management UI, CLI, GMS, or SonicOS API).
- Users with full admin privileges are allowed to access SonicOS API.
- A single SonicOS API session is currently allowed.

For more information refer to API Authentication on page 17.

Examples

Topics:

Example - Commit Pending Configuration

All SonicOS APIs that modify configuration (POST, PUT, DELETE) do not take effect immediately. Rather, configuration is staged and is not pushed to run-time config and saved to flash/permanent storage until API clients explicitly execute a POST request to `/api/sonicos/config/pending`. This is the same behavior as in the SonicOS CLI and equivalent to invoking the `commit` command from the top-level config mode.

Pending configuration can be canceled (deleted) at any time by executing a DELETE request to `/api/sonicos/config/pending`. Any/all pending configuration is canceled upon client session termination, whether due to idle-timeout or explicit logout. In this case, all unsaved changes are lost. It is the client's responsibility to either commit pending configuration after each POST/PUT/DELETE API call or maintain pending changes on the client side to be restored in a later session.

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: `/api/sonicos/config/pending` **Schema:** N/A	Empty	Empty	—	Empty

Topics:

Schema

Schema Structure

A schema is not really applicable here as POST, PUT, and DELETE HTTP body is expected to be empty. However, GET returns any/all pending (unsaved) configuration.

Schema Attributes

Not applicable.

Examples

Topics:

GET Pending Changes (unchanged)

Request:

```
GET /api/sonicos/config/pending
Accept: application/json
```

Response:

```
HTTP/1.0 200 OK
Server: SonicWALL
Content-type: application/json; charset=UTF-8
{
}
```

GET Pending Changes

Request:

```
GET /api/sonicos/config/pending
Accept: application/json
```

Response:

```
HTTP/1.0 200 OK
Server: SonicWALL
Content-type: application/json; charset=UTF-8
{
   "address_objects": [
      {
         "pending": "ADD"
         ,"ipv4": {
            "name": "B"
            ,"host": {
               "ip": "2.2.2.2"
            }
            ,"zone": "WAN"
         }
      }
   ]
}
```

POST Pending Changes

Request:

```
POST /api/sonicos/config/pending
Accept: application/json
```

Response:

```
   HTTP/1.0 200 OK
   Server: SonicWALL
   Content-type: application/json; charset=UTF-8
   {
      "status": {
         "success": true
         ,"cli": {
            "depth": 1
            ,"mode": "config_mode"
            ,"configuring": true
            ,"pending_config": false
            ,"restart_required": "NONE"
         }
         ,"info": [
            { "level": "info", "code": "E_OK", "message": "Success." }
         ]
      }
   }
```

Example - Address Object API Calls

Topics:

Create a new IPv4 Address Object named Web Server

```
POST /api/sonicos/address-objects/ipv4
Content-type: application/json

{
   "address_object": {
      "ipv4": {
         "name": "Web Server",
         "zone": "DMZ",
         "host": {
            "ip": "192.168.168.168"
         }
      }
   }
}
```

Modify the Web Server Address Object host IP

```
PUT /api/sonicos/address-objects/ipv4/name/Web%20Server
Content-type: application/json

{
```

```
   "address_object": {
      "ipv4": {
         "host": {
            "ip": "192.168.168.1"
         }
      }
   }
}
```

Delete the Web Server Address Object

```
DELETE /api/sonicos/address-objects/ipv4/name/Web%20Server
```

2

API Authentication

Topics:

Authentication Methods

SonicOS API supports four authentication mechanisms that share the same endpoint for client login and logout.

Endpoint	HTTP Method & Body			
	GET	POST	PUT	DELETE
URI: `/api/sonicos/auth`	Empty	Empty	—	Empty

1 Navigate to **MANAGE | System Setup | Appliance > Base Settings**.

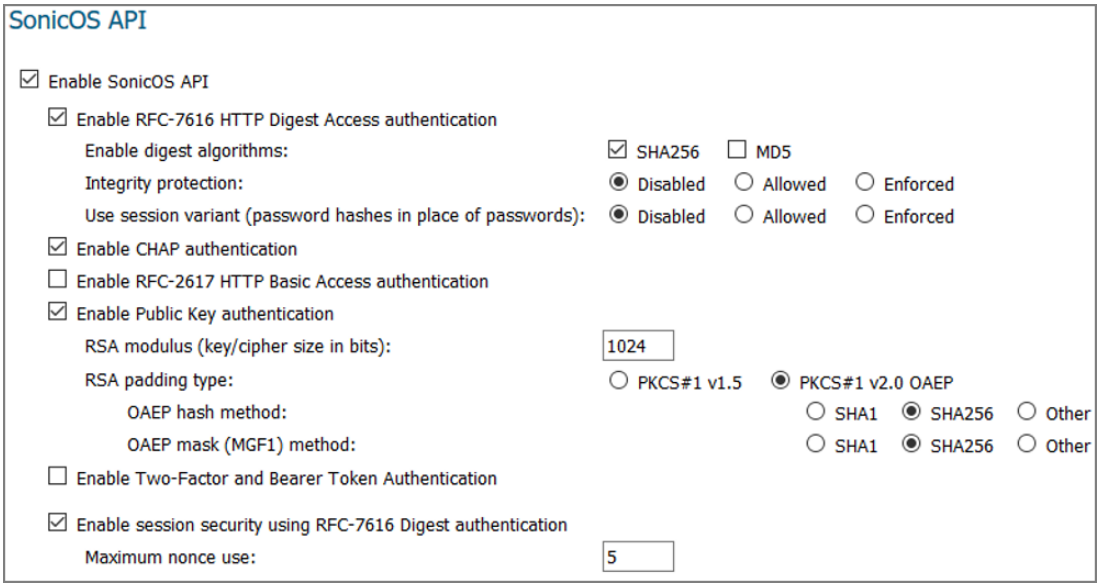
2 Scroll down to the **SonicOS API** section.

3 Select from the choices under **Enable SonicOS API**.

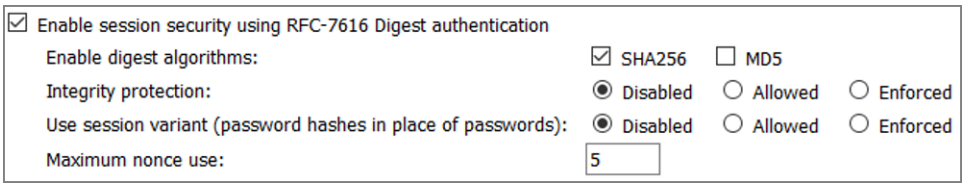
- Enable RFC-7616 HTTP Digest Access Authentication
  - Enable digest algorithms: SHA256 or MD5
  - Integrity protection: Disabled, Allowed, or Enforced.
  - Use session variant (password hashes in place of passwords): Disabled, Allowed, or Enforced.
- Enable CHAP authentication
- Enable RFC-2617 HTTP Basic Access authentication
- Enable Public Key Authentication
  - RSA modulus (key/cipher size in bits): 2014 is the default.
  - RSA padding type: PKCS#1 v1.5 or PKCS#1 v2.0 OAEP
    - OAEP hash method: SHA-1, SHA-256, or Other
    - OAEP mask (MGF1) method: SHA1, SHA-256, or Other
- Enable Two-Factor and Bearer Token Authentication

- Enable session security using RFC-7616 Digest Access Authentication
  - Can hold user passwords received from the client.
  - Maximum nonce use: 10 by default

NOTE: It is highly recommended to call delete api/sonicos/auth to log out of the API session, with bearer token or user name/password. Otherwise, the session is closed after a time of inactivity.



NOTE: The settings for RFC-7616 Digest Authentication also apply to session security. If the settings are disabled for RFC-7616, they are enabled for session security.

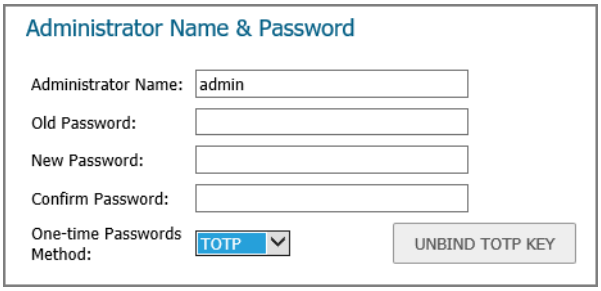


Two-Factor Authentication

SonicOS API supports Two Factor Authentication (TFA) for administrators and users who want to enable the security feature from the Graphical User Interface (GUI) and API. This is an alternative to the other authentication mechanisms described here and cannot be used along with those. Bearer Token Authentication is an alternative method of securing the management requests sent after authentication, as per the Open API Specification, and as used by Swagger. When two-factor authentication is used to log in on the API, then Bearer Token Authentication must be used in all the requests that follow it.

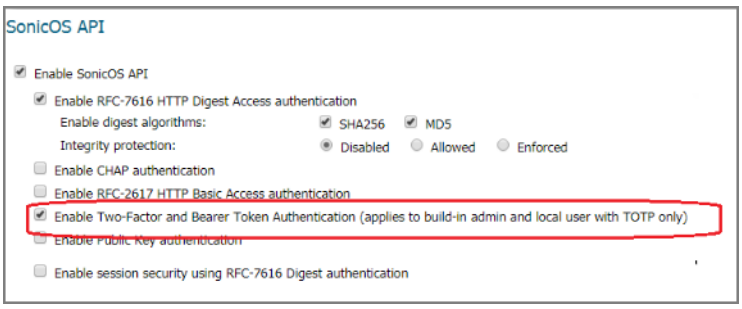
To log in with TFA and use Bearer Token Authentication through the firewall:

1. Enter your **Username** and **Password** in the SonicWall **LOG IN** page.
2. Navigate to **MANAGE | System Setup | Appliance > Base Settings**.
3. Under the **Administrator Name & Password** section, scroll down to **One-time Passwords Method:**
4. Choose **TOTP** from the drop-down menu.



5 Scroll down to the **SonicOS API** section.

6 Select **Enable Two-Factor and Bearer Token Authentication** (applies to built-in admin and local user with TOTP only, post sonicos/tfa directly instead of sonicos/auth).



7 Click **ACCEPT**.

A message displays under the **ACCEPT** and **CANCEL** buttons next to **Status** indicating the configuration has been updated.

To use TFA and Bearer Token Authentication:

1 Enter your **Username** and **Password** in the SonicWall **LOG IN** page.

1 The SonicWall-proprietary bar code screen displays.

2 Install either the **Google Authenticator** or **Duo** apps on your phone to implement two-step verification using TOTP for your appliance.

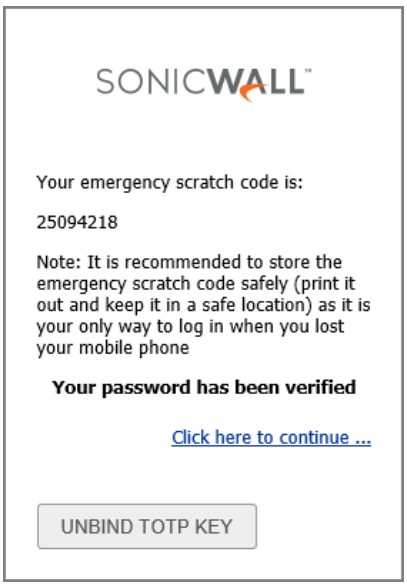
3 Using the apps, scan the SonicWall bar code by positioning your phone lens window in front of the bar code.

4 The apps then generate a security code that you enter into the text field next to **2FA Code:**

IMPORTANT: Remember to write down your eight-digit emergency scratch code somewhere for later access as it is the only way to log in if you lose your mobile phone.

5 Click **OK**.

6 Click the **Click here to continue ...** link in the next SonicWall bar code screen after you have succeeded to **UNBIND** the **TOTP KEY**.



7 Enter the code from the app in the **2FA Code** field and click **OK**.

8 After your password has been verified you successfully land in the appliance's Base Settings page.

NOTE: Administrators and users can also enforce the TFA and Bearer Token Authentication feature by going to **System Setup | Users > Settings** page.

To log in with TFA and use Bearer Token Authentication through the API:

1 Navigate to **MANAGE | Logs & Reporting | API**.

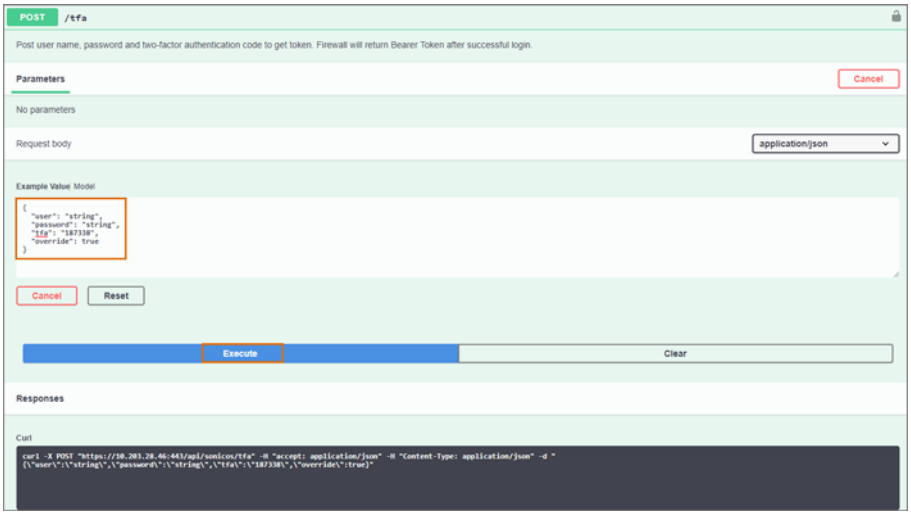
2 Click on the **HTTPS://SONICOS-API.SONICWALL.COM** link under the SonicWall SonicOS API Agreement section.

3 Click **Logout** to log out of the firewall.

4 The browser automatically links to the SWAGGER API open-source software user interface, which displays. You can also use other API tools such as **Postman** and **Linux Command cURL**.

NOTE: The Swagger tool works slowly sometimes so it may take a few seconds for the UI to appear. Also, not all browsers have the same speed of connection to Swagger and the other API apps.

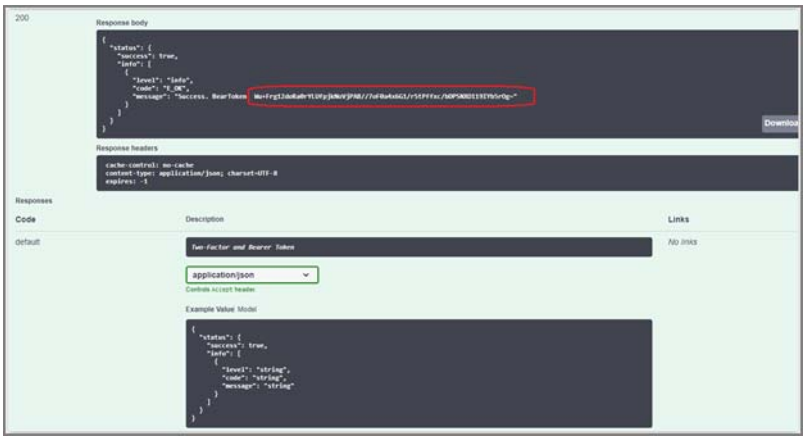
5 Post "**tfa**" with user name, password, and two-factor code to the firewall.



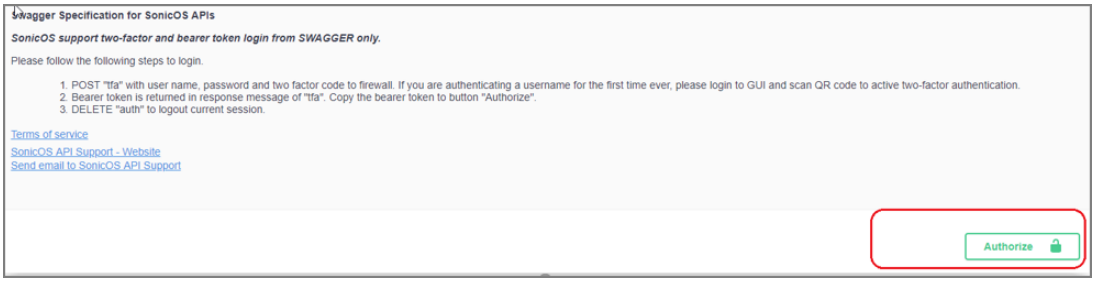
6 Click **Execute**.

7 Click **Authorize** when done.

8 The bearer token is returned in the “tfa” response message.



9 Click **Authorize**.



10 Click **Authorize** again under **Available authorizations**.

RFC-2617 HTTP Basic Authentication

RFC-2617 HTTP Basic Authentication is the simplest method for client authentication. HTTP Basic Authentication uses the standard Authentication HTTP headers to pass user credentials between the client and the server. Because HTTP Basic Authentication provides no means for protecting the confidentiality of a user's credentials, SonicOS API requires user credentials to be transmitted over HTTPS when this is enabled.

For SonicOS API HTTP Basic Authentication, use the Linux command-line `curl` command with the `-u` option:

- Login:

  ```
  curl -k -i -u admin:password -X POST https://a.b.c.d/api/sonicos/auth
  ```

- Logout:

  ```
  curl -k -i -X DELETE https://a.b.c.d/api/sonicos/auth
  ```

RFC-7616 HTTP Digest Access Authentication

SonicOS API supports the RFC-7616 HTTP Digest Access Authentication scheme as its most secure. It includes:

- Secure authentication using SHA-256, extensible for other algorithms in the future.
- Replay prevention utilizing a counter that is incremented in each request and can be reset to any value at any time in replies from the firewall.
- An option for a "rolling nonce," where an HTTP reply can optionally pass back a new nonce (random number) to be used for the next request.
- Optional "integrity protection" where requests with entity body content can include that in the digest calculation.
- An optional "session" variant that uses a SHA hash of the password instead of the password itself so that the SonicWall/client do not need to store the actual password.

For SonicOS API HTTP Digest Access Authentication, use the Linux command-line `curl` command with the `-u` option:

- Login:

  ```
  curl -k -i -u admin:password -digest -X HEAD https://a.b.c.d/api/sonicos/auth
  ```

MD5 Support

MD5 is supported with HTTP Digest Access Authentication to allow inter-operation with older software that does not support SHA-256, but it is disabled by default and use of SHA-256 instead is highly recommended.

SHA-512/256 Support

Although RFC-7616 specifies the hashing scheme named “SHA-512/256,” which is an efficient hybrid between SHA-512 and SHA-256 (see FIPS 180-4), as an alternative to SHA-256, SonicOS does not currently support it.

Integrity Protection

Integrity protection is an optional feature specified in RFC-7616 where the body content of a request is included in the digest hash, hence providing protection against malware trying to change or replace that. This is not useful in the authentication request since no sensitive data is sent, but it is supported for session security and, if enabled, can be used there too.

NOTE: curl’s latest digest authentication does not support integrity protection for requests with data content. Setting integrity protection on the SonicWall to **Allow,** rather than to **Enforce,** allows initial authentication without integrity protection. Custom scripts can then use integrity protection to safeguard the content of the API management requests.

Session Variant

RFC-7616 specifies a mode of operation referred to as the session variant. A hash of the password, and some other fixed values, is used instead of the actual password. This allows the operation without needing to store the password in any retrievable way. This can be useful to enhance security on the client side when using local user accounts, including the built-in admin. The client can then store the hash of the admin password, rather than storing the actual password.

This can also be helpful on the SonicWall side during session security. Refer to Password and Password-Hash Saving on page 29.

Public Key Authentication

The SonicWall proprietary Public Key Authentication is an alternative secure scheme that, unlike digest authentication, allows the password to be securely encrypted and sent from the client to the firewall. This is necessary if session security is to be performed with accounts that are authenticated remotely via LDAP/RADIUS/TACACS+.

Public Key Challenge/Response

The public key exchange utilizes the **WWW-Authenticate** and **Authorization** HTTP headers, compliant with the access authentication framework specified in RFC-7235 section 2, and with their **auth-scheme** specifying **SNWL-PK-AUTH**.

A client must first invoke a challenge from the firewall by making a request to /api/sonicos/auth. Any method can be used for this, but it is suggested that a POST be used if the override parameter is to be set, or otherwise a HEAD request since no request data content is involved. This solicits a response as follows:

```
HTTP/1.0 401 Unauthorized
Server: SonicWall
WWW-Authenticate: SNWL-PK-AUTH type=RSA, key="..."
```

An exception for authentication is with CHAP authenticated via RADIUS, but it is not compatible with then doing session security.

The client will then need to resend the request to /api/sonicos/auth with an Authorization header as follows:

```
Authorization: SNWL-PK-AUTH user="admin", data="..."
```

The content of the key field in the challenge is the RSA public key, in ASN.1 type RSAPublicKey format (see RFC-3447 section A.1.1) and base64-encoded. This is what comes between the BEGIN and END markers in an RSA key in a .pem file, concatenated into a single line (with the BEGIN/END markers not included). For example with this RSA key:

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCdzKnaH+K2kfpHE2U7SDsbZMpd
Qu8vEYIdDlqrQx7BzQpfBGVy5CbTsJn+RiGPNYjtFAL+7Qux4wqc6aOnpWJoY/
BiBmoEKRumBOd2VJBr599yllfqQbXPwQEd9euWTlvaD7G+OhIWFMCnPRIOFkZxwc
1v+Aqq8FY/A/nMYPYwIDAQAB
-----END PUBLIC KEY-----
```

It would be the string “MIGfMA0G...YwIDAQAB”. The client can take this string and save it as a .pem file, enclosed in ----- **BEGIN PUBLIC KEY----- / -----END PUBLIC KEY-----** header/footer. That can then be used with openssl’s command-line RSA utility to encrypt a password using openssl by either of:

echo -n password | openssl rsautl -encrypt -pubin -inkey key-file.pem | base64 -w 0

echo -n password | openssl pkeyutl -encrypt -pkeyopt rsa_padding_mode:pkcs1 -pubin \

-inkey key-file.pem | base64 -w 0

Or if PKCS#1 v2.0 OAEP padding is selected (see below) by any one of:

echo -n password | openssl rsautl -encrypt -oaep -pubin -inkey key-file.pem | base64 -w 0

echo -n password | openssl pkeyutl -encrypt -pkeyopt rsa_padding_mode:oaep -pubin \

-inkey key-file.pem | base64 -w 0

echo -n password | openssl pkeyutl -encrypt -pkeyopt rsa_padding_mode:oaep -pkeyopt \

rsa_oaep_md:sha256 -pkeyopt rsa_mgf1_md:sha256 -pubin _inkey key-file.pem | base64 -w 0

The “openssl rsautl” is now deprecated and using “openssl pkeyutl” is preferred. In the case of OAEP, the first two forms above both use SHA-1 for the hashes in the OAEP padding, and the third form needs to be used for other hashing methods (such as SHA-256, as in the above example). Note that the latter is not supported in all SonicOS version.

The **data** field in the response holds the cipher data (encrypted password), base64-encoded (as output by the above commands).

The string could be piped through "fold -w 64" to break it into 64-character lines, as in the example above, but that is not necessary and a .pem file with a single long line between the header/footer lines works fine.

The following is an example bash script to send a public key authentication request to the firewall, extract the public key from the **WWW-Authenticate** challenge in the reply, use that to encrypt the password (with OAEP padding using SHA-256) and then send the response back to the firewall:

```
curl -k -i -s -X POST https://$ADDR/api/sonicos/auth | grep 'WWW-Authenticate: SNWL-PK-AUTH' \
        | sed -e 's/^.*key="/-----BEGIN PUBLIC KEY-----\n/' \
                -e 's/"/\n-----END PUBLIC KEY-----/' >pk.pem
CIPHER=$(echo -n "$PASSWORD" | openssl pkeyutl -encrypt -pkeyopt rsa_padding_mode:oaep \
        -pkeyopt rsa_oaep_md:sha256 -pkeyopt rsa_mgf1_md:sha256 -pubin -inkey pk.pem \
        | base64 -w 0)
curl -k -i -s -H 'Authorization: SNWL-PK-AUTH user="'$USERNAME'", data="'$CIPHER'"' \
-X POST https://192.168.168.32/api/sonicos/auth
```

RSA Padding

RSA defines two types of padding, the original one specified in **PKCS#1 v1.5**, and a more recent **OAEP** padding specified in **PKCS#1 v2.0**.

PKCS#1 v2.0 utilizes **SHA** hashing and is more secure and preferred, but gives more size overhead, hence resulting in a smaller maximum password size for a given key size. Refer to Password Size Limits and RSA Key Sizes.

The type of padding to use is configurable, defaulting to OAEP. The client and firewall must be using the same type of padding, and for security it is highly recommended that OAEP padding be used.

OAEP padding uses two hashes (its primary hash and that for its **MGF1** mask generation function) and in some versions of SonicOS these too are configurable. In both cases any hashing method that is supported by OpenSSL (the version used in SonicOS) can be used. The two do not need to be the same, but what the client uses in the encryption must match what is configured on the firewall.

Password Size Limits and RSA Key Sizes

The maximum length of the password that can be encrypted depends on the chosen RSA key size (modulus) and padding type, as follows:

Key Bits	Cipher Size	Padding Type	Pad Bytes	Maximum Password Length
512	64 bytes	PKCS#1 v1.5	11	53 characters
512	64 bytes	OAEP with SHA-1	42	22 characters
512	64 bytes	OAEP with SHA-224	58	6 characters
512	64 bytes	OAEP with SHA-256	66	Not possible
512	64 bytes	OAEP with SHA-384	98	Not possible
512	64 bytes	OAEP with SHA-512	130	Not possible
1024	128 bytes	PKCS#1 v1.5	11	117 characters
1024	128 bytes	OAEP with SHA-1	42	86 characters
1024	128 bytes	OAEP with SHA-224	58	70 characters
1024	128 bytes	OAEP with SHA-256	66	62 characters
1024	128 bytes	OAEP with SHA-384	98	30 characters
1024	128 bytes	OAEP with SHA-512	130	Not possible
2048	256 bytes	PKCS#1 v1.5	11	245 characters
2048	256 bytes	OAEP with SHA-1	42	214 characters
2048	256 bytes	OAEP with SHA-224	58	198 characters
2048	256 bytes	OAEP with SHA-256	66	190 characters
2048	256 bytes	OAEP with SHA-384	98	158 characters
2048	256 bytes	OAEP with SHA-512	130	126 characters

Challenge-Handshake Authentication (CHAP)

SonicOS API supports a CHAP authentication scheme, which is generally less secure than the more modern RFC-7616 HTTP Digest scheme, but could be useful, particularly if using RADIUS for the back-end authentication with remote user accounts

Clients must first perform a CHAP challenge initiate request by invoking a call to `GET /api/sonicos/auth`:

```
HTTP/1.0 200 OK
Server: SonicWALL
Content-type: application/json; charset=UTF-8

{
    "id": "{string}",
    "challenge": "{string}"
}
```

id:	Type:	string (hexadecimal number)
	Description:	CHAP ID
	Example:	`0b`

challenge:	Type:	string (hexadecimal #)
	Description:	Hexadecimal-formatted, randomly generated number
	Example:	`EA7F57F37595B6891C222EF284C05D84`

Clients must then generate a one-way hash (CHAP digest) using the user's credentials and the parameters returned via the initiate request. For information on how to calculate the digest see RFC-1994.

When the CHAP digest is generated, it is packaged up via a JSON-formatted request to `POST /api/sonicos/auth`:

```
{
    "override": {boolean},
    "id": "{string}",
    "user": "{string}",
    "digest": "{string}"
}
```

override:	Type:	boolean
	Description:	Boolean flag that if true will allow the API session to override an admin currently logged in.
	Default:	false
	Example:	`true`
id:	Type:	string (hexadecimal number)
	Description:	CHAP ID.
	Example:	`0b`
user:	Type:	string
	Description:	Username.
	Example:	`admin`
digest:	Type:	string
	Description:	CHAP digest.
	Example:	`D96E46E27497B6891C222EF284C05D84`

Pros and Cons of the Different Schemes

Each of the four authentication schemes supported by SonicOS 6.5.4 API has pros and cons, and not all of them are usable in all situations.

Generally, the recommendation is to use Public Key Authentication if administrative user accounts are used that need to be authenticated remotely via RADIUS, LDAP or TACACS+, and use HTTP Digest Authentication otherwise.

Refer to the overview table below for a comparison:

Situations	HTTP Basic	HTTP Digest	Public Key	CHAP
Level of security:	Low	Very High	High	Medium
Supported in 3rd party utilities (curl, etc.):	Yes	Yes	No	No
Client complexity:	Low	Low	Medium-High	Medium
Remote authentication:	Compatible with all	Not possible	Compatible with all	RADIUS only
Efficiency/performance:	High	Medium	Low	Medium

Session Security

Session Security means validating every request that is sent throughout the session after the initial authentication (i.e. those sent to a management, rather than authentication, endpoint). This is to avoid vulnerability to attacks such as injection of malicious requests from malware that can spoof the client's IP address (e.g. cross-site request forgery - CSRF) or a man-in-the-middle attack that could try to alter the content of a request. SonicOS API supports this enforcement which is enabled by default.

For this the RFC-7617 HTTP Digest Access Authentication mechanism is used, which provides for very good session security, including source authentication, replay detection and optional content integrity validation. If session security is enabled on the API then every subsequent management request sent after authentication will need to include an Authorization header generated as per RFC-7617, with an incrementing nc (nonce-count) field.

Session security will be possible after initial authentication by any of the supported schemes, with the one exception that it will not be supported after CHAP authentication with a remote user account authenticated by RADIUS.

Challenge-Free and Challenge/Response Operation

If the client saves the nonce and opaque values from the authentication stage and uses those with a sequential nonce count to generate **Authorization** headers in its requests then, so long as those are valid, no challenge is needed, allowing for efficient operation with a single HTTP request/response for each API management operation. It is recommended that this should be the normal method of operation for most clients.

On the other hand, the client can choose to not do this, sending its requests initially without an **Authorization** header, in which case each request solicits a **401 Unauthorized** response with an HTTP digest challenge to which the client can respond. Operating in this way is less efficient, with two request/response exchanges needed for every API management operation, but it means that a utility like curl, which does not support tracking nonces etc. across multiple requests, can be used without needing additional scripting.

Password and Password-Hash Saving

To perform session security with user accounts that are remotely authenticated via LDAP/RADIUS/TACACS+, the initial authentication must use one of the HTTP Basic Access or Public Key authentication schemes. With these, the client sends the user's password to the SonicWall, and it can then save it for the lifetime of the session and use it for session security validation. If RFC-7617's Session Variant is used then, rather than storing the actual password in its internal memory, the SonicWall stores a more secure irreversible hash of it. The client must then calculate its digest hash accordingly, as per the RFC.

Operation After Non-Digest Authentication

The API client needs to know the values (realm, nonce, opaque and qop) for session security. After initial user authentication by the digest scheme, it already has those and can immediately start sending API requests with digests calculated from them. But if a different mechanism is used for that, the client has two choices:

- The client can send the first request after authentication with no Authorization header, which provokes a digest challenge giving all the relevant data.

- On success of any authentication mechanism, other than HTTP digest authentication, if session security is enabled on the API then the “200 OK” response includes an **Authentication-Info** header giving the data as follows:

  ```
  HTTP/1.0 200 OK
  Server: SonicWALL
  Authentication-Info: Digest algorithm=SHA-256, realm="admin-users@a.b.c.d",
        qop="auth", nonce=”…”, opaque=”…"
  ```

  This follows the model of the Authentication-Info header specified in RFC-7616, but it is a proprietary use of it. Clients can ignore this header (proceeding as per the first bullet option above) but utilizing the data returned in it to avoid the need for the challenge/response handshake when the first post-authentication API management request is sent.

Nonce Resetting

RFC-7617 allows for multiple requests to use the same nonce (with a sequentially updating nonce count) through session, but it also provides a mechanism for the server to periodically (or whenever it chooses) generate a new random nonce, returning it to the client via a **nextnonce** field in an Authentication-Info header in the response to a request. After receiving a response with that, the client must then use it for the next request (resetting the nonce count to 1 for that request).

There is a **Maximum nonce use** configuration option to set the number of requests after which a new nonce is generated. Setting this to zero causes the same nonce to be used through the entire session.

3

API: Config - Pending

About Modifying Configuration API

All SonicOS API that modify configuration (POST, PUT, DELETE) do not take effect immediately. Rather, configuration is staged and is not pushed to run-time config or saved to `flash/permanent` storage until API clients explicitly execute a POST request to `/api/sonicos/config/pending`. This is the same behavior as SonicOS CLI and equivalent to invoking the `commit` command from the top-level config mode.

Pending configuration can be canceled (deleted) at any time by executing a DELETE request to `/api/sonicos/config/pending`. It should be noted that any/all pending configuration is canceled (deleted) upon client session termination, whether due to idle-timeout or explicit logout. In this case, all unsaved changes are lost so it is the client's responsibility to either commit pending configuration after each POST/PUT/DELETE API call or maintain pending changes on the client side to be restored in a later session.

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: `/api/sonicos/config/pending` **Schema:** N/A	Empty	Empty	—	Empty

Schema Structure

A schema is not really applicable here as POST, PUT and DELETE HTTP body is expected to be empty. However, GET returns any/all pending (unsaved) configuration so see all schemas in the following chapters.

Schema Attributes

Not applicable.

Examples

Topics:

GET Pending Changes (Unchanged)

Request:

```
GET /api/sonicos/config/pending
Accept: application/json
```

Response:

```
HTTP/1.0 200 OK
Server: SonicWALL
Content-type: application/json; charset=UTF-8

{

}
```

GET Pending Changes

Request:

```
GET /api/sonicos/config/pending
Accept: application/json
```

Response:

```
HTTP/1.0 200 OK
Server: SonicWALL
Content-type: application/json; charset=UTF-8

{
     "address_objects": [
         {
             "pending": "ADD"
            ,"ipv4": {
                 "name": "B"
                ,"host": {
                     "ip": "2.2.2.2"
                 }
                ,"zone": "WAN"
             }
         }
```

```
    ]
}
```

POST Pending Changes

Request:

```
POST /api/sonicos/config/pending
Accept: application/json
```

Response:

```
HTTP/1.0 200 OK
Server: SonicWALL
Content-type: application/json; charset=UTF-8

{
     "status": {
        "success": true

        ,"cli": {
             "depth": 1
            ,"mode": "config_mode"
            ,"configuring": true
            ,"pending_config": false
            ,"restart_required": "NONE"
        }

        ,"info": [
             { "level": "info", "code": "E_OK", "message": "Success." }
        ]
    }
}
```

4

API: Restart

About Restarting API

Restarts SonicOS (and chassis) immediately or after an interval of time.

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE

URI: `/api/sonicos/restart`
`[ / chassis ]`
`[`
`/at/{YYYYMMDDHHMMSS}`
`| /in/{UINT32} { /minutes | /hours | /days }`
`| /now`
`]`
	—	Empty	—	—

Schema: N/A				

Schema Structure

Not applicable.

Schema Attributes

Not applicable.

Example

```
POST /api/sonicos/restart
POST /api/sonicos/restart/now
POST /api/sonicos/restart/chassis/now
POST /api/sonicos/restart/in/3/days
```

5

API: Address Objects – IPv4

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/address-objects/ipv4* **Schema:** *collection#address-object-ipv4-config*	Empty	Required	Required	Required
URI: */api/sonicos/address-objects/ipv4/name/{NAME}* **Schema:** *object#address-object-ipv4-config*	Empty	—	Required	Ignored
URI: */api/sonicos/address-objects/ipv4/uuid/{UUID}* **Schema:** *object#address-object-ipv4-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: Address Object

```
{
    "address_object": {
        "ipv4": {
            "name": "{string}",
            "uuid": "{string}",

            "host": {
                "ip": "{string}"
            },
```

```
          | "range": {
                "begin": "{string}",
                "end": "{string}"
            },
          | "network": {
                "subnet": "{string}",
                "mask": "{string}"
            }

            "zone": "{string}"
        }
    }
}
```

Collection: Address Object

```
{
    "address_objects": [
        object#address-object-ipv4-config,
        ...
    ]
}
```

Schema Attributes

Topics:

address_object:

```
Type: object
Flags: -none-
Description: Add/edit address object.
```

address_objects:

```
Type: array
Flags: -none-
Description: Address object collection.
```

address_object.ipv4:

```
Type: object
Flags: key
Description: IPV4 address object.
```

address_object.ipv4.name:

```
Type: string
Flags: key
Description: Host/network/range address object name.
```

address_object.ipv4.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

address_object.ipv4.host:

```
Type: object
Flags: -none-
Description: Address object host.
```

address_object.ipv4.host.ip:

```
Type: string (ip)
Flags: -none-
Description: IPv4 host address in the form: D.D.D.D.
```

address_object.ipv4.range:

```
Type: object
Flags: -none-
Description: Address object range.
```

address_object.ipv4.range.begin:

```
Type: string (ip)
Flags: -none-
Description: IPv4 starting range in the form: D.D.D.D.
```

address_object.ipv4.range.end:

```
Type: string (ip)
Flags: -none-
Description: IIPv4 ending range in the form: D.D.D.D.
```

address_object.ipv4.network:

```
Type: object
Flags: -none-
Description: Address object network.
```

address_object.ipv4.network.subnet:

```
Type: string (ip)
Flags: -none-
Description: IPv4 network in the form: D.D.D.D.
```

address_object.ipv4.network.mask:

```
Type: string (subnet)
Flags: -none-
Description: IPv4 netmask in decimal dotted or CIDR form: D.D.D.D OR /D
```

address_object.ipv4.zone:

```
Type: string
Flags: -none-
Description: Zone object name.
```

6

API: Address Objects – IPv6

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/address-objects/ipv6* **Schema:** *collection#address-object-ipv6-config*	Empty	Required	Required	Required
URI: */api/sonicos/address-objects/ipv6/name/{NAME}* **Schema:** *object#address-object-ipv6-config*	Empty	—	Required	Ignored
URI: */api/sonicos/address-objects/ipv6/uuid/{UUID}* **Schema:** *object#address-object-ipv6-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: Address Object

```
{
    "address_object": {
        "ipv6": {
            "name": "{string}",
            "uuid": "{string}",

            "host": {
                "ip": "{string}"
            },
```

```
          | "range": {
                "begin": "{string}",
                "end": "{string}"
            },
          | "network": {
                "subnet": "{string}",
                "mask": "{string}"
            }

            "zone": "{string}"
        }
    }
}
```

Collection: Address Objects

```
{
    "address_objects": [
        object#address-object-ipv6-config,
        ...
    ]
}
```

Schema Attributes

Topics:

address_object:

```
Type: object
Flags: -none-
Description: Add/edit address object.
```

address_objects:

```
Type: array
Flags: -none-
Description: Address object collection.
```

address_object.ipv6:

```
Type: object
Flags: key
Description: IPV6 address object.
```

address_object.ipv6.name:

```
Type: string
Flags: key
Description: Host/network/range address object name.
```

address_object.ipv6.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

address_object.ipv6.host:

```
Type: object
Flags: -none-
Description: Address object host.
```

address_object.ipv6.host.ip:

```
Type: string (ip)
Flags: -none-
Description: IPv6 host address in the form: HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH
```

address_object.ipv6.range:

```
Type: object
Flags: -none-
Description: Address object range.
```

address_object.ipv6.range.begin:

```
Type: string (ip)
Flags: -none-
Description: IPv6 starting range in the form: HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH
```

address_object.ipv6.range.end:

```
Type: string (ip)
Flags: -none-
Description: IIPv6 ending range in the form: HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH
```

address_object.ipv6.network:

```
Type: object
Flags: -none-
Description: Address object network.
```

address_object.ipv6.network.subnet:

```
Type: string (ip)
Flags: -none-
Description: IPv6 network in the form: HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH
```

address_object.ipv6.network.mask:

```
Type: string (v6 prefix)
Flags: -none-
Description: Network prefix.
```

address_object.ipv6.zone:

```
Type: string
Flags: -none-
Description: Zone object name.
```

7

API: Address Objects – MAC

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/address-objects/*mac **Schema:** *collection#address-object-mac-config*	Empty	Required	Required	Required
URI: */api/sonicos/address-objects/mac/name/{NAME}* **Schema:** *object#address-object-mac-config*	Empty	—	Required	Ignored
URI: */api/sonicos/address-objects/mac/uuid/{UUID}* **Schema:** *object#address-object-mac-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: Address Object

```
{
    "address_object": {
        "mac": {
            "name": "{string}",
            "uuid": "{string}",
            "address": "{string}",
            "zone": "{string}",
            "multi_homed": {boolean}
        }
```

```
    }
}
```

Collection: Address Object

```
{
    "address_objects": [
        object#address-object-mac-config,
        ...
    ]
}
```

Schema Attributes

Topics:

address_object:

```
Type: object
Flags: -none-
Description: address object.
```

address_objects:

```
Type: array
Flags: -none-
Description: Address object collection.
```

address_object.mac:

```
Type: object
Flags: key
Description: MAC address object.
```

address_object.mac.name:

```
Type: string
Flags: key
Description: MAC address object name.
```

address_object.mac.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

address_object.mac.address

```
Type: string (mac)
Flags: -none-
Description: Address object MAC address in the form: HH:HH:HH:HH:HH:HH or HHHHHHHHHHHH or
HH-HH-HH-HH-HH-HH.
```

address_object.mac.zone:

```
Type: string
Flags: -none-
Description: Zone object name.
```

address_object.mac.multi_homed:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable multi-homed host.
```

8

API: Address Objects – FQDN

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/address-objects/*fqdn **Schema:** *collection#address-object-fqdn-config*	Empty	Required	Required	Required
URI: */api/sonicos/address-objects/fqdn/name/{NAME}* **Schema:** *object#address-object-fqdn-config*	Empty	—	Required	Ignored
URI: */api/sonicos/address-objects/fqdn/uuid/{UUID}* **Schema:** *object#address-object-fqdn-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: Address Object

```
{
       "address_object": {
        "fqdn": {
            "name": "{string}",
            "uuid": "{string}",
            "domain": "{string}",
            "zone": "{string}",
            "dns_ttl": {number}
        }
```

```
    }
}
```

Collection: Address Object

```
{
    "address_objects": [
        object#address-object-fqdn-config,
        ...
    ]
}
```

Schema Attributes

Topics:

address_object:

```
Type: object
Flags: -none-
Description: address object.
```

address_objects:

```
Type: array
Flags: -none-
Description: Address object collection.
```

address_object.fqdn:

```
Type: object
Flags: key
Description: fqdn address object.
```

address_object.fqdn.name:

```
Type: string
Flags: key
Description: FQDN address object name.
```

address_object.fqdn.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

address_object.fqdn.domain

```
Type: string (fqdn)
Flags: -none-
Description: FQDN in the form: example.com or *.example.com.
```

address_object.fqdn.zone:

```
Type: string
Flags: -none-
Description: Zone object name.
```

address_object.fqdn.dns_ttl

```
Type: number (uint16))
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

9

API: Address Groups — IPv4

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/address-groups/*ipv4 **Schema:** *collection#address-group-ipv4-config*	Empty	Required	Required	Required
URI: */api/sonicos/address-groups/ipv4/name/{NAME}* **Schema:** *object#address-group-ipv4-config*	Empty	—	Required	If deleting member(s)
URI: */api/sonicos/address-groups/ipv4/uuid/{UUID}* **Schema:** *object#address-group-ipv4-config*	Empty	—	Required	If deleting member(s)

Schema Structure

Topics:

Object: Address Group

```
{
    "address_group": {
        "ipv4": {
            "name": "{string}",
            "uuid": "{string}",

            "address_group": {
                "ipv4": [
                    {
                        "name": "{string}"
```

```
                },

                ...
            ]
        },

        "address_object": {
            "ipv4": [
                {
                    "name": "{string}"
                },

                ...
            ],

            "mac": [
                {
                    "name": "{string}"
                },

                ...
            ],

            "fqdn": [
                {
                    "name": "{string}"
                },

                ...
            ]
        }
    }
}
```

Collection: Address Group

```
{
    "address_objects": [
        object#address-group-ipv4-config,
        ...
    ]
}
```

Schema Attributes

Topics:

address_group:

```
Type: object
Flags: -none-
Description: Address group.
```

address_groups:

```
Type: array
Flags: -none-
Description: Address group collection.
```

address_group.ipv4:

```
Type: object
Flags: key
Description: ipv4 address group.
```

address_group.ipv4.name:

```
Type: string
Flags: key
Description: IPv4 address group name.
```

address_group.ipv4.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

address_group.ipv4.address_group:

```
Type: object
Flags: -none-
Description: Assign address group to group.
```

address_group.ipv4.address_group.ipv4:

```
Type: array
Flags: -none-
Description: IPV4 address group.
```

address_group.ipv4.address_group.ipv4.name:

```
Type: string
Flags: -none-
Description: Group address object name.
```

address_group.ipv4.address_object:

```
Type: object
Flags: -none-
Description: Assign an FQDN address object to group.
```

address_group.ipv4.address_object.ipv4:

```
Type: array
Flags: -none-
Description: IPV4 address object.
```

address_group.ipv4.address_object.ipv4.name:

```
Type: string
Flags: -none-
Description: Host/network/range address object name.
```

address_group.ipv4.address_object.mac:

```
Type: array
Flags: -none-
Description: MAC address object.
```

address_group.ipv4.address_object.mac.name:

```
Type: string
Flags: -none-
Description: MAC address object name.
```

address_group.ipv4.address_object.fqdn:

```
Type: array
Flags: -none-
Description: FQDN address object.
```

address_group.ipv4.address_object.fqdn.name:

```
Type: string
Flags: -none-
Description: FQDN address object name.
```

10

API: Address Groups — IPv6

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/address-groups/*ipv6 **Schema:** *collection#address-group-ipv6-config*	Empty	Required	Required	Required
URI: */api/sonicos/address-groups/ipv6/name/{NAME}* **Schema:** *object#address-group-ipv6-config*	Empty	—	Required	If deleting member(s)
URI: */api/sonicos/address-groups/ipv6/uuid/{UUID}* **Schema:** *object#address-group-ipv6-config*	Empty	—	Required	If deleting member(s)

Schema Structure

Topics:

Object: Address Group

```
{
"ipv6": {
            "name": "{string}",
            "uuid": "{string}",

            "address_group": {
                "ipv4": [
                    {
                        "name": "{string}"
                    },
```

```
                ...
            ],

            "ipv6": [
                {
                    "name": "{string}"
                },

                ...
            ]
        },

        "address_object": {
            "ipv4": [
                {
                    "name": "{string}"
                },

                ...
            ],

            "ipv6": [
                {
                    "name": "{string}"
                },

                ...
            ],

            "mac": [
                {
                    "name": "{string}"
                },

                ...
            ],

            "fqdn": [
                {
                    "name": "{string}"
                },

                ...
            ]
        }
    }
}
```

Collection: Address Group

```
{
    "address_objects": [
        object#address-group-ipv6-config,
        ...
    ]
}
```

Schema Attributes

Topics:

address_group:

```
Type: object
Flags: -none-
Description: Address group.
```

address_groups:

```
Type: array
Flags: -none-
Description: Address group collection.
```

address_group.ipv6:

```
Type: object
Flags: key
Description: IPV6 address group.
```

address_group.ipv6.name:

```
Type: string
Flags: key
Description: Group address object name.
```

address_group.ipv6.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

address_group.ipv6.address_group:

```
Type: object
Flags: -none-
Description: Assign address group to group.
```

address_group.ipv6.address_group.ipv4:

```
Type: array
Flags: -none-
Description: IPV4 address group.
```

address_group.ipv6.address_group.ipv4.name:

```
Type: string
Flags: -none-
Description: Group address object name.
```

address_group.ipv6.address_group.ipv6:

```
Type: array
Flags: -none-
Description: IPV6 address group.
```

address_group.ipv6.address_group.ipv6.name:

```
Type: string
Flags: -none-
Description: Group address object name.
```

address_group.ipv6.address_object:

```
Type: object
Flags: -none-
Description: Assign an IPV6 address object to group.
```

address_group.ipv6.address_object.ipv4:

```
Type: array
Flags: -none-
Description: IPV4 address object.
```

address_group.ipv6.address_object.ipv4.name:

```
Type: string
Flags: -none-
Description: Host/network/range address object name.
```

address_group.ipv6.address_object.ipv6:

```
Type: array
Flags: -none-
Description: IPV6 address object.
```

address_group.ipv6.address_object.ipv6.name:

```
Type: string
Flags: -none-
Description: Address object name.
```

address_group.ipv6.address_object.mac:

```
Type: array
Flags: -none-
Description: MAC address object.
```

address_group.ipv6.address_object.mac.name:

```
Type: string
Flags: -none-
Description: MAC address object name.
```

address_group.ipv6.address_object.fqdn:

```
Type: array
Flags: -none-
Description: FQDN address object.
```

address_group.ipv6.address_object.fqdn.name:

```
Type: string
Flags: -none-
Description: FQDN address object name.
```

11

API: Schedule Objects

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/schedules* **Schema:** *collection#schedule-config*	Empty	Required	Required	Required
URI: */api/sonicos/schedules/name/{NAME}* **Schema:** *object#schedule-config*	Empty	—	Required	Ignored
URI: */api/sonicos/schedules/uuid/{UUID}* **Schema:** *object#schedule-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: Schedule

```
{
"schedule": {
        "name": "{string}",
        "uuid": "{string}",

        "occurs": {
            "once": {
                "event": {
                    "start": "{string}",
                    "end": "{string}"
```

```
            }
          },
        | "recurring": {
            "recurring": [
                {
                    "start": "{string}",
                    "end": "{string}",
                    "sun": {boolean},
                    "mon": {boolean},
                    "tue": {boolean},
                    "wed": {boolean},
                    "thu": {boolean},
                    "fri": {boolean},
                    "sat": {boolean}
                },

                ...
            ]
          },
        | "mixed": {
            "event": {
                "start": "{string}",
                "end": "{string}"
            },

            "recurring": [
                {
                    "start": "{string}",
                    "end": "{string}",
                    "sun": {boolean},
                    "mon": {boolean},
                    "tue": {boolean},
                    "wed": {boolean},
                    "thu": {boolean},
                    "fri": {boolean},
                    "sat": {boolean}
                },

                ...
            ]
          }
      }
  }
}
```

Collection: Schedule

```
{
    "schedules": [
        object#schedule-config,
        ...
    ]
}
```

Schema Attributes

Topics:

schedule:

```
Type: object
Flags: -none-
Description: Schedule object.
```

schedules:

```
Type: array
Flags: -none-
Description: Schedule object collection.
```

schedule.name:

```
Type: string
Flags: key
Description:
```

schedule.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

schedule.occurs:

```
Type: object
Flags: -none-
Description: Set schedule type.
```

schedule.occurs.once:

```
Type: object
Flags: -none-
Description: Set for single occurrence.
```

schedule.occurs.once.event:

```
Type: object
Flags: -none-
Description: Enter the start and end date and time of a one time event.
```

schedule.occurs.once.event.start:

```
Type: string (time yyyymmddhhmm)
Flags: -none-
Description: Timestamp in the form: YYYY:MM:DD:HH:MM
```

schedule.occurs.once.event.end:

```
Type: string (time yyyymmddhhmm)
Flags: -none-
Description: Timestamp in the form: YYYY:MM:DD:HH:MM
```

schedule.occurs.recurring:

```
Type: object
Flags: -none-
Description: Set for recurring schedule.
```

schedule.occurs.recurring.recurring:

```
Type: array
Flags: -none-
Description: Add to the list of applicable days and start and stop time of the schedule.
```

schedule.occurs.recurring.recurring.start:

```
Type: string (time hhmm)
Flags: -none-
Description: Time in the form: DD:DD
```

schedule.occurs.recurring.recurring.end:

```
Type: string (time hhmm)
Flags: -none-
Description: Time in the form: DD:DD
```

schedule.occurs.recurring.recurring.sun:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.recurring.recurring.mon:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.recurring.recurring.tue:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.recurring.recurring.wed:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.recurring.recurring.thu:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.recurring.recurring.fri:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.recurring.recurring.sat:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.mixed:

```
Type: object
Flags: -none-
Description: Set for both recurring schedule and single occurrence.
```

schedule.occurs.mixed.event:

```
Type: object
Flags: -none-
Description: Enter the start and end date and time of a one time event.
```

schedule.occurs.mixed.event.start:

```
Type: string (time yyyymmddhhmm)
Flags: -none-
Description: Timestamp in the form: YYYY:MM:DD:HH:MM
```

schedule.occurs.mixed.event.end:

```
Type: string (time yyyymmddhhmm)
Flags: -none-
Description: Timestamp in the form: YYYY:MM:DD:HH:MM
```

schedule.occurs.mixed.recurring:

```
Type: array
Flags: -none-
Description: Add to the list of applicable days and start and stop time of the schedule.
```

schedule.occurs.mixed.recurring.start:

```
Type: string (time hhmm)
Flags: -none-
Description: Time in the form: DD:DD
```

schedule.occurs.mixed.recurring.end:

```
Type: string (time hhmm)
Flags: -none-
Description: Time in the form: DD:DD
```

schedule.occurs.mixed.recurring.sun:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.mixed.recurring.mon:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.mixed.recurring.tue:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.mixed.recurring.wed:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.mixed.recurring.thu:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.mixed.recurring.fri:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

schedule.occurs.mixed.recurring.sat:

```
Type: boolean (true|false)
Flags: -none-
Description: Day of the week.
```

12

API: Service Objects

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/service-objects* **Schema:** *collection#service-object-config*	Empty	Required	Required	Required
URI: */api/sonicos/service-objects/name/{NAME}* **Schema:** *object#service-object-config*	Empty	—	Required	Ignored
URI: */api/sonicos/service-objects/uuid/{UUID}* **Schema:** *object#service-object-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: Service Object

```
{
    "service_object": {
        "name": "{string}",
        "uuid": "{string}",

        "custom": {number},
      | "icmp": "{string}",
      | "igmp": "{string}",
      | "tcp": {
```

```
            "begin": {number},
            "end": {number}
        },
      | "udp": {
            "begin": {number},
            "end": {number}
        },
      | "gre": {true},
      | "esp": {true},
      | "6over4": {true},
      | "ah": {true},
      | "icmpv6": "{string}",
      | "eigrp": {true},
      | "ospf": "{string}",
      | "pim": "{string}",
      | "l2tp": {true},
      | "ipcomp": {true}
    }
}

collection#service-object-config
{
    "service_objects": [
        object#service-object-config,
        ...
    ]
}
```

Collection: Service Object

```
{
    "service-objects": [
        object#service-object-config,
        ...
    ]
}
```

Schema Attributes

Topics:

service_object:

```
Type: object
Flags: -none-
Description: Service object.
```

service_objects:

```
Type: array
Flags: -none-
Description: Service object collection.
```

service_object.name:

```
Type: string
Flags: key
Description: Service object name.
```

service_object.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

service_object.custom:

```
Type: number (uint8)
Flags: -none-
Description: Integer in the form: D OR 0xHH
```

service_object.icmp:

```
Type: string
Flags: -none-
Description: Service object ICMP.
```

service_object.igmp:

```
Type: string
Flags: -none-
Description: Service object IGMP.
```

service_object.tcp:

```
Type: object
Flags: -none-
Description: Service object TCP.
```

service_object.tcp.begin:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

service_object.tcp.end:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

service_object.udp:

```
Type: object
Flags: -none-
Description: Service object UDP.
```

service_object.udp.begin:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

service_object.udp.end:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

service_object.gre:

```
Type: boolean (true)
Flags: -none-
Description: Service object GRE.
```

service_object.esp:

```
Type: boolean (true)
Flags: -none-
Description: Service object ESP.
```

service_object.6over4:

```
Type: boolean (true)
Flags: -none-
Description: Service object 6over4.
```

service_object.ah:

```
Type: boolean (true)
Flags: -none-
Description: Service object AH.
```

service_object.icmpv6:

```
Type: string
Flags: -none-
Description: Service object ICMPV6
```

service_object.eigrp:

```
Type: boolean (true)
Flags: -none-
Description: Service object EIGRP.
```

service_object.ospf:

```
Type: string
Flags: -none-
Description: Service object OSPF.
```

service_object.pim:

```
Type: string
Flags: -none-
Description: Service object PIM.
```

service_object.l2tp:

```
Type: boolean (true)
Flags: -none-
Description: Service object l2tp.
```

service_object.ipcomp:

```
Type: boolean (true)
Flags: -none-
Description: Service object ipcomp.
```

13

API: Service Groups

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/service-groups* **Schema:** *collection#service-group-config*	Empty	Required	Required	Required
URI: */api/sonicos/service-groups/name/{NAME}* **Schema:** *object#service-group-config*	Empty	—	Required	If deleting member(s)
URI: */api/sonicos/service-groups/uuid/{UUID}* **Schema:** *object#service-group-config*	Empty	—	Required	If deleting member(s)

Schema Structure

Topics:

Object: Service Group

```
{
    "service_group": {
        "name": "{string}",
        "uuid": "{string}",

        "service_object": [
            {
                "name": "{string}"
            },
```

```
            ...
        ],

        "service_group": [
            {
                "name": "{string}"
            },

            ...
        ]
    }
}
```

Collection: Service Group

```
{
    "service-groups": [
        object#service-group-config,
        ...
    ]
}
```

Schema Attributes

Topics:

service_group:

```
Type: object
Flags: -none-
Description: Service group.
```

service_groups:

```
Type: array
Flags: -none-
Description: Service group collection.
```

service_group.name:

```
Type: string
Flags: key
Description: Service object group name.
```

service_group.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

service_group.service_object:

```
Type: array
Flags: -none-
Description: Assign service object to group.
```

service_group.service_object.name:

```
Type: string
Flags: -none-
Description: Service object name.
```

service_group.service_group:

```
Type: array
Flags: -none-
Description: Assign service group to group.
```

service_group.service_group.name:

```
Type: string
Flags: -none-
Description: Service object group name.
```

14

API: Zones

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/zones* **Schema:** *collection#zone-config*	Empty	Required	Required	Required
URI: */api/sonicos/zones/name/{NAME}* **Schema:** *object#zone-config*	Empty	—	Required	Ignored
URI: */api/sonicos/zones/uuid/{UUID}* **Schema:** *object#zone-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: Zone

```
{
    "zone": {
        "name": "{string}",
        "uuid": "{string}",

        "security_type": "{string}",
        "interface_trust": {boolean},

        "auto_generate_access_rules": {
            "allow_from_to_equal": {boolean},
```

```
    "allow_from_higher": {boolean},
    "allow_to_lower": {boolean},
    "deny_from_lower": {boolean}
},

"websense_content_filtering": {boolean},

"client": {
    "anti_virus": {boolean},
    "content_filtering": {boolean}
},

"gateway_anti_virus": {boolean},
"intrusion_prevention": {boolean},
"app_control": {boolean},
"anti_spyware": {boolean},
"create_group_vpn": {boolean},
"ssl_control": {boolean},
"sslvpn_access": {boolean},

"wireless": {
    "sslvpn_enforcement": {
        "server": {
            "name": "{string}",
          | "host": "{string}"
        },

        "service": {
            "name": "{string}",

          | "protocol": {
                "name": "{string}",
                "begin": {number},
                "end": {number}
            }
        }
    },

    "wifi_sec_enforcement": {
        "exception_service": {
            "name": "{string}",

          | "protocol": {
                "name": "{string}",
                "begin": {number},
                "end": {number}
            }
        }
    },

    "wifi_sec_for_site_to_site_vpn": {boolean},
    "trust_wpa_traffic_as_wifi_sec": {boolean},
    "only_sonicpoint_traffic": {boolean}
},

"guest_services": {
    "inter_guest": {boolean},

    "bypass": {
        "client": {
            "anti_virus": {boolean},
            "content_filtering": {boolean}
        }
    },

    "external_auth": {
        "client_redirect": "{string}",
```

```
                "web_server": {
                    "protocol": "{string}",
                    "name": "{string}",
                    "port": {number},
                    "timeout": {number}
                },

                "message_auth": {
                    "method": "{string}",
                    "shared_secret": "{string}",
                    "confirm_secret": "{string}"
                },

                "social_network": {
                    "facebook": {boolean},
                    "google": {boolean},
                    "twitter": {boolean}
                },

                "auth_pages": {
                    "login": "{string}",
                    "expiration": "{string}",
                    "timeout": "{string}",
                    "max_sessions": "{string}",
                    "traffic_exceeded": "{string}"
                },

                "web_content": {
                    "redirect": {
                        "use_default": {true},
                      | "custom": "{string}"
                    },

                    "server_down": {
                        "use_default": {true},
                      | "custom": "{string}"
                    }
}
```

Collection: Zone

```
{
    "zones": [
        object#zone-config,
        ...
    ]
}
```

Schema Attributes

Topics:

zone:

```
Type: object
Flags: -none-
Description: Zone object.
```

zones:

```
Type: array
Flags: -none-
Description: Zone object collection.
```

zone.name:

```
Type: string
Flags: key
Description: Zone object name.
```

zone.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

zone.security_type:

```
Type: string
Flags: -none-
Description: Set zone security type.
```

zone.interface_trust:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable allow interface trust.
```

zone.auto_generate_access_rules:

```
Type: object
Flags: -none-
Description: Enable auto generate access rules.
```

zone.auto_generate_access_rules.allow_from_to_equal:

```
Type: boolean (true|false)
Flags: -none-
Description: Allow traffic between zones with the same trust level.
```

zone.auto_generate_access_rules.allow_from_higher:

```
Type: boolean (true|false)
Flags: -none-
Description: Allow traffic from zones with higher trust level.
```

zone.auto_generate_access_rules.allow_to_lower:

```
Type: boolean (true|false)
Flags: -none-
Description: Allow traffic to zones with lower trust level.
```

zone.auto_generate_access_rules.deny_from_lower:

```
Type: boolean (true|false)
Flags: -none-
Description: Deny traffic from zones with lower trust level.
```

zone.websense_content_filtering:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable enforce websense enterprise content filtering service.
```

zone.client:

```
Type: object
Flags: -none-
Description: Client settings
```

zone.client.anti_virus:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable client anti-virus enforcement service.
```

zone.client.content_filtering:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable client content filtering services enforcement service.
```

zone.gateway_anti_virus:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable gateway anti-virus service.
```

zone.intrusion_prevention:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable intrusion prevention service.
```

zone.app_control:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable app control service.
```

zone.anti_spyware:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable anti-spyware service.
```

zone.create_group_vpn:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable automatic creation of group VPN for this zone.
```

zone.ssl_control:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable SSL-Control on this zone.
```

zone.sslvpn_access:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable SSL-VPN access this zone.
```

zone.wireless:

```
Type: object
Flags: -none-
Description: Enter wireless zone configuration mode.
```

zone.wireless.sslvpn_enforcement:

```
Type: object
Flags: -none-
Description: Enable SSLVPN enforcement. Set to null or {} if disabled/unconfigured.
```

zone.wireless.sslvpn_enforcement.server:

```
Type: object
Flags: -none-
Description: Set the SSLVPN server as a named address object.
```

zone.wireless.sslvpn_enforcement.server.name:

```
Type: string
Flags: -none-
Description: Host address object name.
```

zone.wireless.sslvpn_enforcement.server.host:

```
Type: string (ip)
Flags: -none-
Description: IPv4 host address in the form: D.D.D.D. IPv6 host address in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.wireless.sslvpn_enforcement.service:

```
Type: object
Flags: -none-
Description: Set the SSLVPN service as a named service object.
```

zone.wireless.sslvpn_enforcement.service.name:

```
Type: string
Flags: -none-
Description: Service object name.
```

zone.wireless.sslvpn_enforcement.service.protocol:

```
Type: object
Flags: -none-
Description: Set the SSLVPN service as a protocol.
```

zone.wireless.sslvpn_enforcement.service.protocol.name:

```
Type: string
Flags: -none-
Description: Service protocol.
```

zone.wireless.sslvpn_enforcement.service.protocol.begin:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

zone.wireless.sslvpn_enforcement.service.protocol.end:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

zone.wireless.wifi_sec_enforcement:

```
Type: object
Flags: -none-
Description: Enable WiFiSec enforcement.
```

zone.wireless.wifi_sec_enforcement.exception_service:

```
Type: object
Flags: -none-
Description: Specify services that are allowed to bypass wifisec enforcement.
```

zone.wireless.wifi_sec_enforcement.exception_service.name:

```
Type: string
Flags: -none-
Description: Service object name.
```

zone.wireless.wifi_sec_enforcement.exception_service.protocol:

```
Type: object
Flags: -none-
Description: Set the WiFiSec exception service as a protocol.
```

zone.wireless.wifi_sec_enforcement.exception_service.protocol.name:

```
Type: string
Flags: -none-
Description: Service protocol.
```

zone.wireless.wifi_sec_enforcement.exception_service.protocol.begin:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

zone.wireless.wifi_sec_enforcement.exception_service.protocol.end:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

zone.wireless.wifi_sec_for_site_to_site_vpn:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable WiFiSec for site-to-site VPN tunnel traversal.
```

zone.wireless.trust_wpa_traffic_as_wifi_sec:

```
Type: boolean (true|false)
Flags: -none-
Description: Trust WPA / WPA2 traffic as WiFiSec.
```

zone.wireless.only_sonicpoint_traffic:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable only allow traffic generated by a SonicPoint/SonicPointN.
```

zone.guest_services:

```
Type: object
Flags: -none-
Description: Enable zone guest services and enter configuration mode. Set to null or {} if
disabled/unconfigured.
```

zone.guest_services.inter_guest:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable inter-guest communication.
```

zone.guest_services.bypass:

```
Type: object
Flags: -none-
Description: Enable bypass check for guest clients.
```

zone.guest_services.bypass.client:

```
Type: object
Flags: -none-
Description: Enable bypass check for guest clients.
```

zone.guest_services.bypass.client.anti_virus:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable bypass anti-virus check for guests.
```

zone.guest_services.bypass.client.content_filtering:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable bypass client content filtering check for guests.
```

zone.guest_services.external_auth:

```
Type: object
Flags: -none-
Description: Enable external guest authentication and enter its configuration mode. Set to
null or {} if disabled/unconfigured.
```

zone.guest_services.external_auth.client_redirect:

```
Type: string
Flags: -none-
Description: Set local web server settings for client redirect.
```

zone.guest_services.external_auth.web_server:

```
Type: object
Flags: -none-
Description: Configure the external web server settings.
```

zone.guest_services.external_auth.web_server.protocol:

```
Type: string
Flags: -none-
Description: Configure the external web server protocol.
```

zone.guest_services.external_auth.web_server.name:

```
Type: string
Flags: -none-
Description: FQDN/host address object name.
```

zone.guest_services.external_auth.web_server.port:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

zone.guest_services.external_auth.web_server.timeout:

```
Type: number (uint8)
Flags: -none-
Description: Integer in the form: D OR 0xHH
```

zone.guest_services.external_auth.message_auth:

```
Type: object
Flags: -none-
Description: Enable external message authentication.
```

zone.guest_services.external_auth.message_auth.method:

```
Type: string
Flags: -none-
Description: Set external message authentication method.
```

zone.guest_services.external_auth.message_auth.shared_secret:

```
Type: string
Flags: -none-
Description:
```

zone.guest_services.external_auth.message_auth.confirm_secret:

```
Type: string
Flags: -none-
Description:
```

zone.guest_services.external_auth.social_network:

```
Type: object
Flags: -none-
Description: Enable social network login.
```

zone.guest_services.external_auth.social_network.facebook:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable Facebook social network login.
```

zone.guest_services.external_auth.social_network.google:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable Google social network login.
```

zone.guest_services.external_auth.social_network.twitter:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable Twitter social network login.
```

zone.guest_services.external_auth.auth_pages:

```
Type: object
Flags: -none-
Description: Configure the external authentication pages.
```

zone.guest_services.external_auth.auth_pages.login:

```
Type: string
Flags: -none-
Description:
```

zone.guest_services.external_auth.auth_pages.expiration:

```
Type: string
Flags: -none-
Description:
```

zone.guest_services.external_auth.auth_pages.timeout:

```
Type: string
Flags: -none-
Description:
```

zone.guest_services.external_auth.auth_pages.max_sessions:

```
Type: string
Flags: -none-
Description:
```

zone.guest_services.external_auth.auth_pages.traffic_exceeded:

```
Type: string
Flags: -none-
Description:
```

zone.guest_services.external_auth.web_content:

```
Type: object
Flags: -none-
Description: Configure the Web content messages.
```

zone.guest_services.external_auth.web_content.redirect:

```
Type: object
Flags: -none-
Description: Configure the Web content redirect message.
```

zone.guest_services.external_auth.web_content.redirect.use_default:

```
Type: boolean (true)
Flags: -none-
Description: Use the default Web content redirect message.
```

zone.guest_services.external_auth.web_content.redirect.custom:

```
Type: string
Flags: -none-
Description:
```

zone.guest_services.external_auth.web_content.server_down:

```
Type: object
Flags: -none-
Description: Configure the Web content server down message.
```

zone.guest_services.external_auth.web_content.server_down.use_default:

```
Type: boolean (true)
Flags: -none-
Description: Use the default Web content server down message.
```

zone.guest_services.external_auth.web_content.server_down.custom:

```
Type: string
Flags: -none-
Description:
```

zone.guest_services.external_auth.logout_expired:

```
Type: object
Flags: -none-
Description: Enable auto-session logout.
```

zone.guest_services.external_auth.logout_expired.every:

```
Type: number (uint8)
Flags: -none-
Description: Integer in the form: D OR 0xHH
```

zone.guest_services.external_auth.logout_expired.cgi:

```
Type: string
Flags: -none-
Description:
```

zone.guest_services.external_auth.status_check:

```
Type: object
Flags: -none-
Description: Enable server status check.
```

zone.guest_services.external_auth.status_check.every:

```
Type: number (uint8)
```

```
   Flags: -none-
   Description: Integer in the form: D OR 0xHH
```

zone.guest_services.external_auth.status_check.cgi:

```
   Type: string
   Flags: -none-
   Description:
```

zone.guest_services.external_auth.session_sync:

```
   Type: object
   Flags: -none-
   Description: Enable session synchronization.
```

zone.guest_services.external_auth.session_sync.every:

```
   Type: number (uint8)
   Flags: -none-
   Description: Integer in the form: D OR 0xHH
```

zone.guest_services.external_auth.session_sync.cgi:

```
   Type: string
   Flags: -none-
   Description:
```

zone.guest_services.policy_page_non_authentication:

```
   Type: object
   Flags: -none-
   Description: Enable policy page without authentication and enter its configuration mode. Set
to null or {} if disabled/unconfigured.
```

zone.guest_services.policy_page_non_authentication.guest_usage_policy:

```
   Type: string
   Flags: -none-
   Description:
```

zone.guest_services.custom_auth_page:

```
   Type: object
   Flags: -none-
   Description: Enable custom authentication page and enter its configuration mode. Set to null
or {} if    disabled/unconfigured.
```

zone.guest_services.custom_auth_page.header:

```
   Type: object
   Flags: -none-
   Description: Configure custom page header.
```

zone.guest_services.custom_auth_page.header.text:

```
   Type: string
```

```
Flags: -none-
Description:
```

zone.guest_services.custom_auth_page.header.url:

```
Type: string (web url)
Flags: -none-
Description: URL in the form: http://host/file
```

zone.guest_services.custom_auth_page.footer:

```
Type: object
Flags: -none-
Description: Configure custom login page footer.
```

zone.guest_services.custom_auth_page.footer.text:

```
Type: string
Flags: -none-
Description:
```

zone.guest_services.custom_auth_page.footer.url:

```
Type: string (web url)
Flags: -none-
Description: URL in the form: http://host/file
```

zone.guest_services.post_auth:

```
Type: string (web url)
Flags: -none-
Description: URL in the form: http://host/file
```

zone.guest_services.bypass_guest_auth:

```
Type: object
Flags: -none-
Description: Enable bypass guest authentication. Set to null or {} if disabled/unconfigured.
```

zone.guest_services.bypass_guest_auth.all:

```
Type: boolean (true)
Flags: -none-
Description: All MAC addresses.
```

zone.guest_services.bypass_guest_auth.name:

```
Type: string
Flags: -none-
Description: MAC address object name.
```

zone.guest_services.bypass_guest_auth.group:

```
Type: string
Flags: -none-
Description: MAC group address object name.
```

zone.guest_services.bypass_guest_auth.mac:

```
   Type: string (mac)
   Flags: -none-
   Description: Address object MAC address in the form: HH:HH:HH:HH:HH:HH or HHHHHHHHHHHH or
HH-HH-HH-HH-HH-HH.
```

zone.guest_services.smtp_redirect:

```
   Type: object
   Flags: -none-
   Description: Redirect SMTP traffic to specified server. Set to null or {} if
disabled/unconfigured.
```

zone.guest_services.smtp_redirect.name:

```
   Type: string
   Flags: -none-
   Description: Host address object name.
```

zone.guest_services.smtp_redirect.host:

```
   Type: string (ip)
   Flags: -none-
   Description: IPv4 host address in the form: D.D.D.D. IPv6 host address in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.deny_networks:

```
   Type: object
   Flags: -none-
   Description: Enable blocking of traffic to the named network.
```

zone.guest_services.deny_networks.name:

```
   Type: string
   Flags: -none-
   Description: Address object name.
```

zone.guest_services.deny_networks.group:

```
   Type: string
   Flags: -none-
   Description: Group address object name.
```

zone.guest_services.deny_networks.mac:

```
   Type: string (mac)
   Flags: -none-
   Description: Address object MAC address in the form: HH:HH:HH:HH:HH:HH or HHHHHHHHHHHH or
HH-HH-HH-HH-HH-HH.
```

zone.guest_services.deny_networks.fqdn:

```
   Type: string (fqdn)
   Flags: -none-
   Description: FQDN in the form: example.com or *.example.com.
```

zone.guest_services.deny_networks.host:

```
   Type: string (ip)
   Flags: -none-
   Description: IPv4 host address in the form: D.D.D.D. IPv6 host address in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.deny_networks.range:

```
   Type: object
   Flags: -none-
   Description: Set the denied networks to range of addresses.
```

zone.guest_services.deny_networks.range.begin:

```
   Type: string (ip)
   Flags: -none-
   Description: IPv4 starting range in the form: D.D.D.D. IPv6 starting range in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.deny_networks.range.end:

```
   Type: string (ip)
   Flags: -none-
   Description: IPv4 ending range in the form: D.D.D.D. IPv6 ending range in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.deny_networks.network:

```
   Type: object
   Flags: -none-
   Description: Set the denied networks to network address.
```

zone.guest_services.deny_networks.network.subnet:

```
   Type: string (ip)
   Flags: -none-
   Description: IPv4 network in the form: D.D.D.D. IPv6 network in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.deny_networks.network.mask:

```
   Type: string (subnet)
   Flags: -none-
   Description: IPv4 netmask in decimal dotted or CIDR form: D.D.D.D OR /D. IPv6 netmask in the
form: /D.
```

zone.guest_services.deny_networks.ipv6:

```
   Type: object
   Flags: -none-
   Description: IPv6 address object.
```

zone.guest_services.deny_networks.ipv6.host:

```
   Type: string (ip)
   Flags: -none-
   Description: IPv4 host address in the form: D.D.D.D. IPv6 host address in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.deny_networks.ipv6.range:

```
Type: object
Flags: -none-
Description: Set the denied networks to range of addresses.
```

zone.guest_services.deny_networks.ipv6.range.begin:

```
Type: string (ip)
Flags: -none-
Description: IPv4 starting range in the form: D.D.D.D. IPv6 starting range in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.deny_networks.ipv6.range.end:

```
Type: string (ip)
Flags: -none-
Description: IPv4 ending range in the form: D.D.D.D. IPv6 ending range in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.deny_networks.ipv6.network:

```
Type: object
Flags: -none-
Description: Set the denied networks to network address.
```

zone.guest_services.deny_networks.ipv6.network.subnet:

```
Type: string (ip)
Flags: -none-
Description: IPv4 network in the form: D.D.D.D. IPv6 network in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.deny_networks.ipv6.network.mask:

```
Type: string (subnet)
Flags: -none-
Description: IPv4 netmask in decimal dotted or CIDR form: D.D.D.D OR /D. IPv6 netmask in the
form: /D.
```

zone.guest_services.pass_networks:

```
Type: object
Flags: -none-
Description: Enable allowing of traffic to the named network.
```

zone.guest_services.pass_networks.name:

```
Type: string
Flags: -none-
Description: Address object name.
```

zone.guest_services.pass_networks.group:

```
Type: string
Flags: -none-
Description: Group address object name.
```

zone.guest_services.pass_networks.mac:

```
   Type: string (mac)
   Flags: -none-
   Description: Address object MAC address in the form: HH:HH:HH:HH:HH:HH or HHHHHHHHHHHH or
HH-HH-HH-HH-HH-HH.
```

zone.guest_services.pass_networks.fqdn:

```
   Type: string (fqdn)
   Flags: -none-
   Description: FQDN in the form: example.com or *.example.com.
```

zone.guest_services.pass_networks.host:

```
   Type: string (ip)
   Flags: -none-
   Description: IPv4 host address in the form: D.D.D.D. IPv6 host address in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.pass_networks.range:

```
   Type: object
   Flags: -none-
   Description: Set the pass networks to range of addresses.
```

zone.guest_services.pass_networks.range.begin:

```
   Type: string (ip)
   Flags: -none-
   Description: IPv4 starting range in the form: D.D.D.D. IPv6 starting range in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.pass_networks.range.end:

```
   Type: string (ip)
   Flags: -none-
   Description: IPv4 ending range in the form: D.D.D.D. IPv6 ending range in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.pass_networks.network:

```
   Type: object
   Flags: -none-
   Description: Set the pass networks to network address.
```

zone.guest_services.pass_networks.network.subnet:

```
   Type: string (ip)
   Flags: -none-
   Description: IPv4 network in the form: D.D.D.D. IPv6 network in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.pass_networks.network.mask:

```
   Type: string (subnet)
   Flags: -none-
   Description: IPv4 netmask in decimal dotted or CIDR form: D.D.D.D OR /D. IPv6 netmask in the
form: /D.
```

zone.guest_services.pass_networks.ipv6:

```
Type: object
Flags: -none-
Description: IPv6 address object.
```

zone.guest_services.pass_networks.ipv6.host:

```
Type: string (ip)
Flags: -none-
Description: IPv4 host address in the form: D.D.D.D. IPv6 host address in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.pass_networks.ipv6.range:

```
Type: object
Flags: -none-
Description: Set the pass networks to range of addresses.
```

zone.guest_services.pass_networks.ipv6.range.begin:

```
Type: string (ip)
Flags: -none-
Description: IPv4 starting range in the form: D.D.D.D. IPv6 starting range in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.pass_networks.ipv6.range.end:

```
Type: string (ip)
Flags: -none-
Description: IPv4 ending range in the form: D.D.D.D. IPv6 ending range in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.pass_networks.ipv6.network:

```
Type: object
Flags: -none-
Description: Set the pass networks to network address.
```

zone.guest_services.pass_networks.ipv6.network.subnet:

```
Type: string (ip)
Flags: -none-
Description: IPv4 network in the form: D.D.D.D. IPv6 network in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

zone.guest_services.pass_networks.ipv6.network.mask:

```
Type: string (subnet)
Flags: -none-
Description: IPv4 netmask in decimal dotted or CIDR form: D.D.D.D OR /D. IPv6 netmask in the
form: /D.
```

zone.guest_services.max_guests:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

zone.guest_services.dynamic_address_translation:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable dynamic address translation.
```

15

API: DNS

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/dns* **Schema:** *collection#dns-config*	Empty	—	Required	—

Schema Structure

Topics:

Object: DNS

```
{
    "dns": {
        "server": {
            "inherit": {true},
          | "static": {
                "primary": "{string}",
                "secondary": "{string}",
                "tertiary": "{string}"
            }

            "ipv6": {
                "preferred": {boolean},

                "inherit": {true},
              | "static": {
                    "primary": "{string}",
                    "secondary": "{string}",
                    "tertiary": "{string}"
                }
```

```
            }
        },

        "rebinding": {
            "action": "{string}",

            "allowed_domains": {
                "name": "{string}",
              | "group": "{string}"
            }
        },

        "fqdn_binding": {boolean}
    }
}
```

Schema Attributes

Topics:

dns:

```
Type: object
Flags: -none-
Description: DNS configuration.
```

dns.server:

```
Type: object
Flags: -none-
Description: DNS server configuration.
```

dns.server.inherit:

```
Type: boolean (true)
Flags: -none-
Description: Inherit DNS servers.
```

dns.server.static:

```
Type: object
Flags: -none-
Description: Set static DNS server
```

dns.server.static.primary:

```
Type: string (ip)
Flags: -none-
Description: IPv4 host address in the form: D.D.D.D
```

dns.server.static.secondary:

```
Type: string (ip)
Flags: -none-
Description: IPv4 host address in the form: D.D.D.D
```

dns.server.static.tertiary:

```
Type: string (ip)
Flags: -none-
Description: IPv4 host address in the form: D.D.D.D
```

dns.server.ipv6:

```
Type: object
Flags: -none-
Description: Set IPv6 DNS server
```

dns.server.ipv6.preferred:

```
Type: boolean
Flags: -none-
Description: Prefer IPv6 DNS servers.
```

dns.server.ipv6.inherit:

```
Type: boolean (true)
Flags: -none-
Description: Inherit DNS servers.
```

dns.server.ipv6.static:

```
Type: object
Flags: -none-
Description: Set static DNS server
```

dns.server.ipv6.static.primary:

```
Type: string (ip)
Flags: -none-
Description: IIPv6 host address in the form: HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH
```

dns.server.ipv6.static.secondary:

```
Type: string (ip)
Flags: -none-
Description: IPv6 host address in the form: HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH
```

dns.server.ipv6.static.tertiary:

```
Type: string (ip)
Flags: -none-
Description: IPv6 host address in the form: HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH
```

dns.rebinding:

```
Type: object
Flags: -none-
Description: Enable and configure DNS rebinding attack prevention. Set to null or {} if
disabled/unconfigured.
```

Disabling rebinding example:

```
{
    "dns": {
        "rebinding": {
        }
    }
}
```

dns.rebinding.action:

```
Type: string
Flags: -none-
Description: Set action when experiencing attack. Must be one of the following values:
log-attack-only | return-query-refused | drop-dns-reply
```

dns.rebinding.allowed_domains:

```
Type: object
Flags: -none-
Description: Specify the domains for which checking is not done. Set to null or {} if
disabled/unconfigured.
```

Disabling allowed_domains example:

```
{
    "dns": {
        "rebinding": {
```

```
            "allowed_domains": {
            }
        }
    }
}
```

dns.rebinding.allowed_domains.name:

```
Type: string
Flags: -none-
Description: FQDN address object name.
```

dns.rebinding.allowed_domains.group:

```
Type: string
Flags: -none-
Description: Custom FQDN group address object name.
```

dns.fqdn_binding:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable FQDN object only cache DNS reply from sanctioned server.
```

16

API: Interfaces – IPv4

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/interfaces/ipv4* **Schema:** *collection#interface-ipv4-config*	Empty	—	Required	—
URI: */api/sonicos/interfaces/ipv4* **Schema:** *collection#interface-ipv4-config*	Empty	—	Required	—

Schema Structure

Topics:

Object: Interface – IPv4

```
{
    "interface": {
        "ipv4": {
            "name": "{string}",
            "comment": "{string}",

            "ip_assignment": {
                "zone": "{string}",

                "mode": {
                    "static": {
                        "ip": "{string}",
                        "netmask": "{string}",
```

```
            "gateway": "{string}",

            "dns": {
                "primary": "{string}",
                "secondary": "{string}",
                "tertiary": "{string}"
            },

            "backup_ip": "{string}"
        },

      | "dhcp": {
            "hostname": "{string}",
            "renew_on_startup": {boolean},
            "renew_on_link_up": {boolean},
            "initiate_renewals_with_discover": {boolean},
            "force_discover_interval": {number}
        }
    }
},

"mtu": {number},

"mac": {
    "default": {true},
  | "override": "{string}"
},

"link_speed": {
    "auto_negotiate": {true},
  | "half": "{string}",
  | "full": "{string}"
},

"management": {
    "http": {boolean},
    "https": {boolean},
    "ping": {boolean},
    "snmp": {boolean},
    "ssh": {boolean}
},

"user_login": {
    "http": {boolean},
    "https": {boolean}
},

"https_redirect": {boolean},
"send_icmp_fragmentation": {boolean},
"fragment_packets": {boolean},
"ignore_df_bit": {boolean},
"flow_reporting": {boolean},
"multicast": {boolean},
"cos_8021p": {boolean},
"exclude_route": {boolean},
"asymmetric_route": {boolean},
"shutdown_port": {boolean},
"default_8021p_cos": "{string}",
"policy": "{string}",

"sonicpoint": {
    "limit": {number},

    "reserve_address": {
        "dynamic": {true},
      | "manual": "{string}"
    }
```

```
            }
        }
    }
}
```

Collection: Interface – IPv4

```
{
    "interfaces": [
        object#interface-ipv4-config,
        ...
    ]
}
```

Schema Attributes

Topics:

interface:

```
Type: object
Flags: -none-
Description: Interface.
```

interfaces:

```
Type: array
Flags: -none-
Description: Interface collection.
```

interface.ipv4:

```
Type: object
Flags: -none-
Description: IP version IPV4.
```

interface.ipv4.name:

```
Type: string
Flags: key
Description: Interface name.
```

interface.ipv4.comment:

```
Type: string
Flags: -none-
Description:
```

interface.ipv4.ip_assignment:

```
Type: object
Flags: -none-
Description: Set interface zone and IP assignment. Set to null or {} if
disabled/unconfigured.
```

interface.ipv4.ip_assignment.zone:

```
Type: string
Flags: -none-
Description: Zone object name.
```

interface.ipv4.ip_assignment.mode:

```
Type: object
Flags: -none-
Description: Interface IP assignment mode.
```

interface.ipv4.ip_assignment.mode.static:

```
Type: object
Flags: -none-
Description: Static IP address assignment.
```

interface.ipv4.ip_assignment.mode.static.ip:

```
Type: string (v4 ip)
Flags: -none-
Description: IPV4 Address in the form: a.b.c.d
```

interface.ipv4.ip_assignment.mode.static.netmask:

```
Type: string (v4 subnet)
Flags: -none-
Description: IPV4 netmask in decimal dotted or CIDR form: D.D.D.D OR /D
```

interface.ipv4.ip_assignment.mode.static.gateway:

```
Type: string (v4 ip)
Flags: -none-
Description: IPV4 Address in the form: a.b.c.d
```

interface.ipv4.ip_assignment.mode.static.dns:

```
Type: object
Flags: -none-
Description: Set the DNS server IP address.
```

interface.ipv4.ip_assignment.mode.static.dns.primary:

```
Type: string (v4 ip)
Flags: -none-
Description: IPV4 Address in the form: a.b.c.d
```

interface.ipv4.ip_assignment.mode.static.dns.secondary:

```
Type: string (v4 ip)
Flags: -none-
Description: IPV4 Address in the form: a.b.c.d
```

interface.ipv4.ip_assignment.mode.static.dns.tertiary:

```
Type: string (v4 ip)
Flags: -none-
Description: IPV4 Address in the form: a.b.c.d
```

interface.ipv4.ip_assignment.mode.static.backup_ip:

```
Type: string (v4 ip)
Flags: -none-
Description: IPV4 Address in the form: a.b.c.d
```

interface.ipv4.ip_assignment.mode.dhcp:

```
Type: object
Flags: -none-
Description: IP address obtained by DHCP.
```

interface.ipv4.ip_assignment.mode.dhcp.hostname:

```
Type: string
Flags: -none-
Description:
```

interface.ipv4.ip_assignment.mode.dhcp.renew_on_startup:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable request renew of previous IP on startup.
```

interface.ipv4.ip_assignment.mode.dhcp.renew_on_link_up:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable renew DHCP lease on any link up occurrence.
```

interface.ipv4.ip_assignment.mode.dhcp.initiate_renewals_with_discover:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable initiate renewals with a discover when using DHCP.
```

interface.ipv4.ip_assignment.mode.dhcp.force_discover_interval:

```
Type: number (uint32)
Flags: -none-
Description: Integer in the form: D OR 0xHHHHHHHH
```

interface.ipv4.mtu:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

interface.ipv4.mac:

```
Type: object
Flags: -none-
Description: Set MAC address used for this interface.
```

interface.ipv4.mac.default:

```
Type: boolean (true)
Flags: -none-
Description: Factory configured MAC.
```

interface.ipv4.mac.override:

```
Type: string (mac)
Flags: -none-
Description: MAC address in the form: HH:HH:HH:HH:HH:HH OR HHHHHHHHHHHH
```

interface.ipv4.link_speed:

```
Type: object
```

```
Flags: -none-
Description: Set interface link speed.
```

interface.ipv4.link_speed.auto_negotiate:

```
Type: boolean (true)
Flags: -none-
Description: Set interface link speed to auto-negotiate.
```

interface.ipv4.link_speed.half:

```
Type: string
Flags: -none-
Description: Half duplex.
```

interface.ipv4.link_speed.full:

```
Type: string
Flags: -none-
Description: Full duplex.
```

interface.ipv4.management:

```
Type: object
Flags: -none-
Description: Enable management for the specified protocols.
```

interface.ipv4.management.http:

```
Type: boolean (true|false)
Flags: -none-
Description: HTTP.
```

interface.ipv4.management.https:

```
Type: boolean (true|false)
Flags: -none-
Description: HTTPS.
```

interface.ipv4.management.ping:

```
Type: boolean (true|false)
Flags: -none-
Description: Ping.
```

interface.ipv4.management.snmp:

```
Type: boolean (true|false)
Flags: -none-
Description: SNMP.
```

interface.ipv4.management.ssh:

```
Type: boolean (true|false)
Flags: -none-
Description: SSH.
```

interface.ipv4.user_login:

```
Type: object
Flags: -none-
Description: Enable user login for the specified protocols.
```

interface.ipv4.user_login.http:

```
Type: boolean (true|false)
Flags: -none-
Description: HTTP.
```

interface.ipv4.user_login.https:

```
Type: boolean (true|false)
Flags: -none-
Description: HTTPS.
```

interface.ipv4.https_redirect:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable redirection from HTTP to HTTPS.
```

interface.ipv4.send_icmp_fragmentation:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable ICMP fragmentation needed message generation.
```

interface.ipv4.fragment_packets:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable fragment non-VPN outbound packets larger than this interface's MTU.
```

interface.ipv4.ignore_df_bit:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable ignore don't fragment (DF) bit.
```

interface.ipv4.flow_reporting:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable flow reporting on the interface.
```

interface.ipv4.multicast:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable multicast support.
```

interface.ipv4.cos_8021p:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable 802.1p support.
```

interface.ipv4.exclude_route:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable exclude from route advertisement (NSM, OSPF, BGP, RIP).
```

interface.ipv4.asymmetric_route:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable asymmetric route.
```

interface.ipv4.shutdown_port:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable shutdown port.
```

interface.ipv4.default_8021p_cos:

```
Type: string
Flags: -none-
Description: Enable default 802.1p CoS.
```

interface.ipv4.policy:

```
Type: string
Flags: -none-
Description: Tunnel interface VPN policy name.
```

interface.ipv4.sonicpoint:

```
Type: object
Flags: -none-
Description: Set SonicPoint parameter.
```

interface.ipv4.sonicpoint.limit:

```
Type: number (uint32)
Flags: -none-
Description: SonicPoint limit per interface.
```

interface.ipv4.sonicpoint.reserve_address:

```
Type: object
Flags: -none-
Description: Set dynamically or manually reserve SonicPoint address.
```

interface.ipv4.sonicpoint.reserve_address.dynamic:

```
Type: boolean (true)
Flags: -none-
Description: Dynamically reserve SonicPoint address.
```

interface.ipv4.sonicpoint.reserve_address.manual:

```
Type: string (v4 ip)
Flags: -none-
Description: IPV4 Address in the form: a.b.c.d
```

17

API: NAT Policies – IPv4

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/nat-policies/ipv4* **Schema:** *collection#nat-policies-ipv4-config*	Empty	Required	Required	Required
URI: */api/sonicos/nat-policies/ipv4* **Schema:** *collection#nat-policies-ipv4-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: NAT Policies – IPv4

```
{
    "nat_policy": {
        "ipv4": {
            "inbound": "{string}",
            "outbound": "{string}",

            "source": {
                "any": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "translated_source": {
```

```
            "original": {true},
          | "name": "{string}",
          | "group": "{string}"
        },

        "destination": {
            "any": {true},
          | "name": "{string}",
          | "group": "{string}"
        },

        "translated_destination": {
            "original": {true},
          | "name": "{string}",
          | "group": "{string}"
        },

        "service": {
            "any": {true},
          | "name": "{string}",
          | "group": "{string}"
        },

        "translated_service": {
            "original": {true},
          | "name": "{string}",
          | "group": "{string}"
        },

        "uuid": "{string}",
        "name": "{string}",
        "enable": {boolean},
        "comment": "{string}",

        "priority": {
            "auto": {true},
          | "manual": {number}
        },

        "reflexive": {boolean},

        "virtual_group": {
            "any": {true},
          | "id": {number}
        },

        "nat_method": "{string}",
        "source_port_remap": {boolean},

        "high_availability": {
            "probing": {
                "probe_every": {number},

                "probe_type": {
                    "icmp_ping": {true},
                  | "tcp": {number}
                },

                "reply_timeout": {number},
                "deactivate_after": {number},
                "reactivate_after": {number},
                "rst_as_miss": {boolean},
                "port_probing": {boolean}
            }
        }
    }
```

```
    }
}
```

Collection: NAT Policies – IPv4

```
{
    "nat_policies": [
        object#nat-policy-ipv4-config,
        ...
    ]
}
```

Schema Attributes

Topics:

nat_policy:

```
Type: object
Flags: -none-
Description: NAT policy.
```

nat_policies:

```
Type: array
Flags: -none-
Description: NAT policy collection.
```

nat_policy.ipv4:

```
Type: object
Flags: -none-
Description: IPv4 NAT policy.
```

nat_policy.ipv4.inbound:

```
Type: string
Flags: key
Description: Interface name.
```

nat_policy.ipv4.outbound:

```
Type: string
Flags: key
Description: Interface name.
```

nat_policy.ipv4.source:

```
Type: object
Flags: key
Description: Specify the original source for the NAT policy.
```

nat_policy.ipv4.source.any:

```
Type: boolean (true)
Flags: key
Description: Any host.
```

nat_policy.ipv4.source.name:

```
Type: string
Flags: key
Description: Address object name.
```

nat_policy.ipv4.source.group:

```
Type: string
Flags: key
Description: Group address object name.
```

nat_policy.ipv4.translated_source:

```
Type: object
Flags: key
Description: Specify the translated source for the NAT policy.
```

nat_policy.ipv4.translated_source.original:

```
Type: boolean (true)
Flags: key
Description: Original source IP.
```

nat_policy.ipv4.translated_source.name:

```
Type: string
Flags: key
Description: Address object name.
```

nat_policy.ipv4.translated_source.group:

```
Type: string
Flags: key
Description: Group address object name.
```

nat_policy.ipv4.destination:

```
Type: object
Flags: key
Description: Specify the original destination for the NAT policy.
```

nat_policy.ipv4.destination.any:

```
Type: boolean (true)
Flags: key
Description: Any host.
```

nat_policy.ipv4.destination.name:

```
Type: string
Flags: key
Description: Address object name.
```

nat_policy.ipv4.destination.group:

```
Type: string
Flags: key
Description: Group address object name.
```

nat_policy.ipv4.translated_destination:

```
Type: object
Flags: key
Description: Specify the translated destination for the NAT policy.
```

nat_policy.ipv4.translated_destination.original:

```
Type: boolean (true)
Flags: key
Description: Original destination IP.
```

nat_policy.ipv4.translated_destination.name:

```
Type: string
Flags: key
Description: Address object name.
```

nat_policy.ipv4.translated_destination.group:

```
Type: string
Flags: key
Description: Group address object name.
```

nat_policy.ipv4.service:

```
Type: object
Flags: key
Description: Specify the original service for the NAT policy.
```

nat_policy.ipv4.service.any:

```
Type: boolean (true)
Flags: key
Description: Any service.
```

nat_policy.ipv4.service.name:

```
Type: string
Flags: key
Description: Service object name.
```

nat_policy.ipv4.service.group:

```
Type: string
Flags: key
Description: Service object group name.
```

nat_policy.ipv4.translated_service:

```
Type: object
Flags: key
Description: Specify the translated service for the NAT policy.
```

nat_policy.ipv4.translated_service.original:

```
Type: boolean (true)
Flags: key
Description: Original service.
```

nat_policy.ipv4.translated_service.name:

```
Type: string
Flags: key
Description: Service object name.
```

nat_policy.ipv4.translated_service.group:

```
Type: string
Flags: key
Description: Service object group name.
```

nat_policy.ipv4.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

nat_policy.ipv4.name:

```
Type: string
Flags: required
Description: Name.
```

nat_policy.ipv4.enable:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable NAT policy.
```

nat_policy.ipv4.comment:

```
Type: string
Flags: -none-
Description:
```

nat_policy.ipv4.priority:

```
Type: object
Flags: -none-
Description: Set NAT policy priority
```

nat_policy.ipv4.priority.auto:

```
Type: boolean (true)
Flags: -none-
Description: Set auto priority(priority = 0) for NAT policy.
```

nat_policy.ipv4.priority.manual:

```
Type: number (uint32)
Flags: -none-
Description: Integer in the form: D OR 0xHHHHHHHH
```

nat_policy.ipv4.reflexive:

```
Type: boolean (true|false)
Flags: -none-
Description: Configure a reflexive rule.
```

nat_policy.ipv4.virtual_group:

```
Type: object
Flags: -none-
Description: Specify virtual group for the NAT policy.
```

nat_policy.ipv4.virtual_group.any:

```
Type: boolean (true)
Flags: -none-
Description: Any virtual group.
```

nat_policy.ipv4.virtual_group.id:

```
Type: number (uint8)
Flags: -none-
Description: Integer in the form: D OR 0xHH
```

nat_policy.ipv4.nat_method:

```
Type: string
Flags: -none-
Description: Set the NAT destination translation method.
```

nat_policy.ipv4.source_port_remap:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable source port remap.
```

nat_policy.ipv4.high_availability:

```
Type: object
Flags: -none-
Description: NAT high availability and load balancing configuration mode.
```

nat_policy.ipv4.high_availability.probing:

```
Type: object
Flags: -none-
Description: Enable HA probing and enter configuration mode.
```

nat_policy.ipv4.high_availability.probing.probe_every:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

nat_policy.ipv4.high_availability.probing.probe_type:

```
Type: object
Flags: -none-
Description: Set probe IP type.
```

nat_policy.ipv4.high_availability.probing.probe_type.icmp_ping:

```
Type: boolean (true)
Flags: -none-
Description: ICMP ping probe.
```

nat_policy.ipv4.high_availability.probing.probe_type.tcp:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

nat_policy.ipv4.high_availability.probing.reply_timeout:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

nat_policy.ipv4.high_availability.probing.deactivate_after:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

nat_policy.ipv4.high_availability.probing.reactivate_after:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

nat_policy.ipv4.high_availability.probing.rst_as_miss:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable count RST response as miss.
```

nat_policy.ipv4.high_availability.probing.port_probing:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable port probing.
```

18

API: NAT Policies – IPv6

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/nat-policies/ipv6* **Schema:** *collection#nat-policies-ipv6-config*	Empty	Required	Required	Required
URI: */api/sonicos/nat-policies/ipv6* **Schema:** *collection#nat-policies-ipv6-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: NAT Policies – IPv6

```
{
    "nat_policy": {
        "ipv6": {
            "inbound": "{string}",
            "outbound": "{string}",

            "source": {
                "any": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "translated_source": {
                "original": {true},
              | "name": "{string}",
              | "group": "{string}"
```

```
            },

            "destination": {
                "any": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "translated_destination": {
                "original": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "service": {
                "any": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "translated_service": {
                "original": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "uuid": "{string}",
            "name": "{string}",
            "enable": {boolean},
            "comment": "{string}",

            "priority": {
                "auto": {true},
              | "manual": {number}
            },

            "reflexive": {boolean},

            "virtual_group": {
                "any": {true},
              | "id": {number}
            }
        }
    }
}
```

Schema Attributes

Topics:

nat_policy:

```
Type: object
Flags: -none-
Description: NAT policy.
```

nat_policies:

```
Type: object
Flags: -none-
Description: NAT policy collection.
```

nat_policy.ipv6:

```
Type: object
Flags: key
Description: IPv6 NAT policy.
```

nat_policy.ipv6.inbound:

```
Type: string
Flags: key
Description: Interface name.
```

nat_policy.ipv6.outbound:

```
Type: string
Flags: key
Description: Interface name.
```

nat_policy.ipv6.source:

```
Type: object
Flags: key
Description: Specify the original source for the NAT policy.
```

nat_policy.ipv6.source.any:

```
Type: boolean (true)
Flags: key
Description: Any host.
```

nat_policy.ipv6.source.name:

```
Type: string
Flags: key
Description: Address object name.
```

nat_policy.ipv6.source.group:

```
Type: string
Flags: key
Description: Group address object name.
```

nat_policy.ipv6.translated_source:

```
Type: object
Flags: key
Description: Specify the translated source for the NAT policy.
```

nat_policy.ipv6.translated_source.original:

```
Type: boolean (true)
Flags: key
Description: Original source IP.
```

nat_policy.ipv6.translated_source.name:

```
Type: string
Flags: key
Description: Address object name.
```

nat_policy.ipv6.translated_source.group:

```
Type: string
Flags: key
Description: Group address object name.
```

nat_policy.ipv6.destination:

```
Type: object
Flags: key
Description: Specify the original destination for the NAT policy.
```

nat_policy.ipv6.destination.any:

```
Type: boolean (true)
Flags: key
Description: Any host.
```

nat_policy.ipv6.destination.name:

```
Type: string
Flags: key
Description: Address object name.
```

nat_policy.ipv6.destination.group:

```
Type: string
Flags: key
Description: Group address object name.
```

nat_policy.ipv6.translated_destination:

```
Type: object
Flags: key
Description: Specify the translated destination for the NAT policy.
```

nat_policy.ipv6.translated_destination.original:

```
Type: boolean (true)
Flags: key
Description: Original destination IP.
```

nat_policy.ipv6.translated_destination.name:

```
Type: string
Flags: key
Description: Address object name.
```

nat_policy.ipv6.translated_destination.group:

```
Type: string
Flags: key
Description: Group address object name.
```

nat_policy.ipv6.service:

```
Type: object
Flags: key
Description: Specify the original service for the NAT policy.
```

nat_policy.ipv6.service.any:

```
Type: boolean (true)
Flags: key
Description: Any service.
```

nat_policy.ipv6.service.name:

```
Type: string
Flags: key
Description: Service object name.
```

nat_policy.ipv6.service.group:

```
Type: string
Flags: key
Description: Service object group name.
```

nat_policy.ipv6.translated_service:

```
Type: object
Flags: key
Description: Specify the translated service for the NAT policy.
```

nat_policy.ipv6.translated_service.original:

```
Type: boolean (true)
Flags: key
Description: Original service.
```

nat_policy.ipv6.translated_service.name:

```
Type: string
Flags: key
Description: Service object name.
```

nat_policy.ipv6.translated_service.group:

```
Type: string
Flags: key
```

```
Description: Service object group name.
```

nat_policy.ipv6.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

nat_policy.ipv6.name:

```
Type: string
Flags: required
Description: Name.
```

nat_policy.ipv6.enable:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable NAT policy.
```

nat_policy.ipv6.comment:

```
Type: string
Flags: -none-
Description: Policy comment.
```

nat_policy.ipv6.priority:

```
Type: object
Flags: -none-
Description: Set NAT policy priority.
```

nat_policy.ipv6.priority.auto:

```
Type: boolean (true)
Flags: -none-
Description: Set auto priority(priority = 0) for NAT policy.
```

nat_policy.ipv6.priority.manual:

```
Type: number (uint32)
Flags: -none-
Description: Integer in the form: D OR 0xHHHHHHHH
```

nat_policy.ipv6.reflexive:

```
Type: boolean (true|false)
Flags: -none-
Description: Configure a reflexive rule.
```

nat_policy.ipv6.virtual_group:

```
Type: object
Flags: -none-
Description: Specify virtual group for the NAT policy.
```

nat_policy.ipv6.virtual_group.any:

```
Type: boolean (true)
Flags: -none-
Description: Any virtual group.
```

nat_policy.ipv6.virtual_group.id:

```
Type: number (uint8)
Flags: -none-
Description: Integer in the form: D OR 0xHH
```

19

API: NAT Policies – NAT64

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/nat-policies/nat64* **Schema:** *collection#nat-policy-nat64-config*	Empty	Required	Required	Required
URI: */api/sonicos/nat-policies/nat64* **Schema:** *collection#nat-policy-nat64-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: NAT Policies – NAT64

```
{
    "nat_policy": {
        "nat64": {
            "inbound": "{string}",
            "outbound": "{string}",

            "source": {
                "any": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "translated_source": {
```

```
                "original": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "pref64": {
                "any": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "translated_destination": {
                "embedded_ipv4_address": {true}
            },

            "service": {
                "icmp_udp_tcp": {true}
            },

            "translated_service": {
                "original": {true}
            },

            "uuid": "{string}",
            "name": "{string}",
            "enable": {boolean},
            "comment": "{string}"
        }
    }
}
```

Collection: NAT Policies – NAT64

```
{
    "nat_policies": [
        object#nat-policy-nat64-config,
        ...
    ]
}
```

Schema Attributes

Topics:

nat_policy:

```
Type: object
Flags: -none-
Description: NAT policy.
```

nat_policies:

```
Type: object
Flags: -none-
Description: NAT policy collection.
```

nat_policy.nat64:

```
Type: object
Flags: key
Description: NAT64 NAT policy.
```

nat_policy.nat64.inbound:

```
Type: string
Flags: key
Description: Interface name.
```

nat_policy.nat64.outbound:

```
Type: string
Flags: key
Description: Interface name.
```

nat_policy.nat64.source:

```
Type: object
Flags: key
Description: Specify the original source for the NAT64 policy.
```

nat_policy.nat64.source.any:

```
Type: boolean (true)
Flags: key
Description: Any host.
```

nat_policy.nat64.source.name:

```
Type: string
Flags: key
Description: Address object name.
```

nat_policy.nat64.source.group:

```
Type: string
Flags: key
Description: Group address object name.
```

nat_policy.nat64.translated_source:

```
Type: object
Flags: key
Description: Specify the translated source for the NAT64 policy.
```

nat_policy.nat64.translated_source.original:

```
Type: boolean (true)
Flags: key
Description: Original source IP.
```

nat_policy.nat64.translated_source.name:

```
Type: string
Flags: key
Description: Address object name.
```

nat_policy.nat64.translated_source.group:

```
Type: string
Flags: key
Description: Group address object name.
```

nat_policy.nat64.pref64:

```
Type: object
Flags: key
Description: Specify the prefix for the NAT64 policy.
```

nat_policy.nat64.pref64.any:

```
Type: boolean (true)
Flags: key
Description: Any host.
```

nat_policy.nat64.pref64.name:

```
Type: string
Flags: key
Description: Address object name.
```

nat_policy.nat64.pref64.group:

```
Type: string
Flags: key
Description: Group address object name.
```

nat_policy.nat64.translated_destination:

```
Type: object
Flags: key
Description: Specify the translated destination for the NAT policy.
```

nat_policy.nat64.translated_destination.embedded_ipv4_address:

```
Type: boolean (true)
Flags: key
Description: Embedded ipv4 address.
```

nat_policy.nat64.service:

```
Type: object
Flags: key
Description: Specify the original service for the NAT policy.
```

nat_policy.nat64.service.icmp_udp_tcp:

```
Type: boolean (true)
Flags: key
Description: ICMP UDP TCP service.
```

nat_policy.nat64.translated_service:

```
Type: object
Flags: key
Description: Specify the translated service for the NAT policy.
```

nat_policy.nat64.translated_service.original:

```
Type: boolean (true)
Flags: key
Description: Original service.
```

nat_policy.nat64.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

nat_policy.nat64.name:

```
Type: string
Flags: required
Description: Name.
```

nat_policy.nat64.enable:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable NAT policy.
```

nat_policy.nat64.comment:

```
Type: string
Flags: -none-
Description:
```

20

API: Access Rules – IPv4

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/access-rules/ipv4* **Schema:** *collection#access-rule-ipv4-config*	Empty	Required	Required	Required
URI: */api/sonicos/access-rules/ipv4/uuid/{UUID}* **Schema:** *collection#access-rule-ipv4-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: Access Rules – IPv4

```
{
    "access_rule": {
        "ipv4": {
            "from": "{string}",
            "to": "{string}",
            "action": "{string}",

            "source": {
                "address": {
                    "any": {true},
                  | "name": "{string}",
                  | "group": "{string}"
                },
```

```
    "port": {
        "any": {true},
      | "name": "{string}",
      | "group": "{string}"
    }
},

"service": {
    "any": {true},
  | "name": "{string}",
  | "group": "{string}"
},

"destination": {
    "address": {
        "any": {true},
      | "name": "{string}",
      | "group": "{string}"
    }
},

"schedule": {
    "always_on": {true},
  | "name": "{string}"
},

"users": {
    "included": {
        "all": {true},
      | "guests": {true},
      | "administrator": {true},
      | "name": "{string}",
      | "group": "{string}"
    },

    "excluded": {
        "none": {true},
      | "guests": {true},
      | "administrator": {true},
      | "name": "{string}",
      | "group": "{string}"
    }
},

"uuid": "{string}",
"name": "{string}",
"comment": "{string}",
"enable": {boolean},
"reflexive": {boolean},
"max_connections": {number},
"logging": {boolean},
"management": {boolean},
"packet_monitoring": {boolean},

"priority": {
    "auto": {true},
  | "manual": {number}
},

"tcp": {
    "timeout": {number}
},

"udp": {
    "timeout": {number}
},
```

```
            "fragments": {boolean},
            "botnet_filter": {boolean},

            "connection_limit": {
                "destination": {
                    "threshold": {number}
                },

                "source": {
                    "threshold": {number}
                }
            },

            "flow_reporting": {boolean},
            "geo_ip_filter": {boolean},
            "single_sign_on": {boolean},
            "cos_override": {boolean},

            "quality_of_service": {
                "class_of_service": {
                    "explicit": "{string}",
                  | "map": {true},
                  | "preserve": {true}
                },

                "dscp": {
                    "explicit": {number},
                  | "map": {true},
                  | "preserve": {true}
                }
            }
        }
    }
}
```

Collection: Access Rules – IPv4

```
{
    "access_rules": [
        object#access_rule-ipv4-config,
        ...
    ]
}
```

Schema Attributes

Topics:

access_rule:

```
Type: object
Flags: -none-
Description: Access rule.
```

access_rules:

```
Type: array
Flags: -none-
Description: Access rule collection.
```

access_rule.ipv4:

```
Type: object
Flags: -none-
Description: IPv4 access rule.
```

access_rule.ipv4.from:

```
Type: string
Flags: key
Description: Zone object name.
```

access_rule.ipv4.to:

```
Type: string
Flags: key
Description: Zone object name.
```

access_rule.ipv4.action:

```
Type: string
Flags: key
Description: Set the action for this access rule.
```

access_rule.ipv4.source:

```
Type: object
Flags: key
Description: Source.
```

access_rule.ipv4.source.address:

```
Type: object
Flags: key
Description: Source address.
```

access_rule.ipv4.source.address.any:

```
Type: boolean (true)
Flags: key
Description: Any address.
```

access_rule.ipv4.source.address.name:

```
Type: string
Flags: key
Description: Address object name.
```

access_rule.ipv4.source.address.group:

```
Type: string
Flags: key
Description: Group address object name.
```

access_rule.ipv4.source.port:

```
Type: object
Flags: key
Description: Specify a source port for this Access Policy.
```

access_rule.ipv4.source.port.any:

```
Type: boolean (true)
Flags: key
Description: Any source service.
```

access_rule.ipv4.source.port.name:

```
Type: string
Flags: key
Description: Service object name.
```

access_rule.ipv4.source.port.group:

```
Type: string
Flags: key
Description: Service object group name.
```

access_rule.ipv4.service:

```
Type: object
Flags: key
Description: Specify a destination service for this Access Policy.
```

access_rule.ipv4.service.any:

```
Type: boolean (true)
Flags: key
Description: Any destination service.
```

access_rule.ipv4.service.name:

```
Type: string
Flags: key
Description: Service object name.
```

access_rule.ipv4.service.group:

```
Type: string
Flags: key
Description: Service object group name.
```

access_rule.ipv4.destination:

```
Type: object
Flags: key
Description: Destination.
```

access_rule.ipv4.destination.address:

```
Type: object
Flags: key
Description: Destination a destination address for this Access Policy.
```

access_rule.ipv4.destination.address.any:

```
Type: boolean (true)
Flags: key
Description: Any address.
```

access_rule.ipv4.destination.address.name:

```
Type: string
Flags: key
Description: Address object name.
```

access_rule.ipv4.destination.address.group:

```
Type: string
Flags: key
Description: Group address object name.
```

access_rule.ipv4.schedule:

```
Type: object
Flags: key
Description: Specify a schedule for this access policy.
```

access_rule.ipv4.schedule.always_on:

```
Type: boolean (true)
Flags: key
Description: Always on.
```

access_rule.ipv4.schedule.name:

```
Type: string
Flags: key
Description: Schedule object name.
```

access_rule.ipv4.users:

```
Type: object
Flags: key
Description: Specify users that are excluded from this access policy.
```

access_rule.ipv4.users.included:

```
Type: object
Flags: key
Description: Specify included users.
```

access_rule.ipv4.users.included.all:

```
Type: boolean (true)
Flags: key
Description: All users.
```

access_rule.ipv4.users.included.guests:

```
Type: boolean (true)
Flags: key
Description: Guest users.
```

access_rule.ipv4.users.included.administrator:

```
Type: boolean (true)
Flags: key
Description: Administrator.
```

access_rule.ipv4.users.included.name:

```
Type: string
Flags: key
Description: Local user object name.
```

access_rule.ipv4.users.included.group:

```
Type: string
Flags: key
Description: Local user group object name.
```

access_rule.ipv4.users.excluded:

```
Type: object
Flags: key
Description: Specify excluded users.
```

access_rule.ipv4.users.excluded.none:

```
Type: boolean (true)
Flags: key
Description: No users.
```

access_rule.ipv4.users.excluded.guests:

```
Type: boolean (true)
Flags: key
Description: Guest users.
```

access_rule.ipv4.users.excluded.administrator:

```
Type: boolean (true)
Flags: key
Description: Administrator.
```

access_rule.ipv4.users.excluded.name:

```
Type: string
Flags: key
Description: Local user object name.
```

access_rule.ipv4.users.excluded.group:

```
Type: string
Flags: key
Description: Local user group object name.
```

access_rule.ipv4.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

access_rule.ipv4.name:

```
Type: string
Flags: required
Description: Name.
```

access_rule.ipv4.comment:

```
Type: string
Flags: -none-
Description:
```

access_rule.ipv4.enable:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable this access rule.
```

access_rule.ipv4.reflexive:

```
Type: boolean (true|false)
Flags: -none-
Description: Configure a reflexive rule.
```

access_rule.ipv4.max_connections:

```
Type: number (uint8)
Flags: -none-
Description: Integer in the form: D OR 0xHH
```

access_rule.ipv4.logging:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable logging when this access rule is used.
```

access_rule.ipv4.management:

```
Type: boolean (true|false)
Flags: -none-
Description: Allow management traffic.
```

access_rule.ipv4.packet_monitoring:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable packet monitoring.
```

access_rule.ipv4.priority:

```
Type: object
Flags: -none-
Description: Set access rule priority
```

access_rule.ipv4.priority.auto:

```
Type: boolean (true)
Flags: -none-
Description: Set auto priority(priority = 0) for access rule.
```

access_rule.ipv4.priority.manual:

```
Type: number (uint32)
Flags: -none-
Description: Integer in the form: D OR 0xHHHHHHHH
```

access_rule.ipv4.tcp:

```
Type: object
Flags: -none-
Description: TCP.
```

access_rule.ipv4.tcp.timeout:

```
Type: number (uint32)
Flags: -none-
Description: Integer in the form: D OR 0xHHHHHHHH
```

access_rule.ipv4.udp:

```
Type: object
Flags: -none-
Description: UDP.
```

access_rule.ipv4.udp.timeout:

```
Type: number (uint32)
Flags: -none-
Description: Integer in the form: D OR 0xHHHHHHHH
```

access_rule.ipv4.fragments:

```
Type: boolean (true|false)
Flags: -none-
Description: Allow fragmented packets on this access rule.
```

access_rule.ipv4.botnet_filter:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable Botnet filter.
```

access_rule.ipv4.connection_limit:

```
Type: object
Flags: -none-
Description: Configure connection limit.
```

access_rule.ipv4.connection_limit.destination:

```
Type: object
Flags: -none-
Description: Enable connection limit for each destination IP address. Set to null or {} if
disabled/unconfigured.
```

access_rule.ipv4.connection_limit.destination.threshold:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

access_rule.ipv4.connection_limit.source:

```
Type: object
Flags: -none-
Description: Enable connection limit for each source IP address. Set to null or {} if
disabled/unconfigured.
```

access_rule.ipv4.connection_limit.source.threshold:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

access_rule.ipv4.flow_reporting:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable flow reporting.
```

access_rule.ipv4.geo_ip_filter:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable Geo-IP filter.
```

access_rule.ipv4.single_sign_on:

```
Type: boolean (true|false)
Flags: -none-
Description: Invoke single sign on to authenticate users.
```

access_rule.ipv4.cos_override:

```
Type: boolean (true|false)
Flags: -none-
Description: Allow 802.1p marking to override DSCP values.
```

access_rule.ipv4.quality_of_service:

```
Type: object
Flags: -none-
Description: Configure quality of service for rule.
```

access_rule.ipv4.quality_of_service.class_of_service:

```
Type: object
Flags: -none-
Description: Set 802.1p marking action. Set to null or {} if disabled/unconfigured.
```

access_rule.ipv4.quality_of_service.class_of_service.explicit:

```
Type: string
Flags: -none-
Description: Set explicit marking.
```

access_rule.ipv4.quality_of_service.class_of_service.map:

```
Type: boolean (true)
Flags: -none-
Description: Map marking.
```

access_rule.ipv4.quality_of_service.class_of_service.preserve:

```
Type: boolean (true)
Flags: -none-
Description: Preserve marking.
```

access_rule.ipv4.quality_of_service.dscp:

```
Type: object
Flags: -none-
Description: Set DSCP marking action.
```

access_rule.ipv4.quality_of_service.dscp.explicit:

```
Type: number (uint8)
Flags: -none-
Description: Integer in the form: D OR 0xHH
```

access_rule.ipv4.quality_of_service.dscp.map:

```
Type: boolean (true)
Flags: -none-
Description: Map marking.
```

access_rule.ipv4.quality_of_service.dscp.preserve:

```
Type: boolean (true)
Flags: -none-
Description: Preserve marking.
```

21

API: Access Rules – IPv6

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/access-rules/ipv6* **Schema:** *collection#access-rule-ipv6-config*	Empty	Required	Required	Required
URI: */api/sonicos/access-rules/ipv6/uuid/{UUID}* **Schema:** *collection#access-rule-ipv6-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: Access Rules – IPv6

```
{
    "access_rule": {
        "ipv6": {
            "from": "{string}",
            "to": "{string}",
            "action": "{string}",

            "source": {
                "address": {
                    "any": {true},
                  | "name": "{string}",
                  | "group": "{string}"
                },
```

```
    "port": {
        "any": {true},
      | "name": "{string}",
      | "group": "{string}"
    }
},

"service": {
    "any": {true},
  | "name": "{string}",
  | "group": "{string}"
},

"destination": {
    "address": {
        "any": {true},
      | "name": "{string}",
      | "group": "{string}"
    }
},

"schedule": {
    "always_on": {true},
  | "name": "{string}"
},

"users": {
    "included": {
        "all": {true},
      | "guests": {true},
      | "administrator": {true},
      | "name": "{string}",
      | "group": "{string}"
    },

    "excluded": {
        "none": {true},
      | "guests": {true},
      | "administrator": {true},
      | "name": "{string}",
      | "group": "{string}"
    }
},

"uuid": "{string}",
"name": "{string}",
"comment": "{string}",
"enable": {boolean},
"reflexive": {boolean},
"max_connections": {number},
"logging": {boolean},
"management": {boolean},
"packet_monitoring": {boolean},

"priority": {
    "auto": {true},
  | "manual": {number}
},

"tcp": {
    "timeout": {number}
},

"udp": {
    "timeout": {number}
},
```

```
            "fragments": {boolean},
            "botnet_filter": {boolean},

            "connection_limit": {
                "destination": {
                    "threshold": {number}
                },

                "source": {
                    "threshold": {number}
                }
            },

            "flow_reporting": {boolean},
            "geo_ip_filter": {boolean},
            "single_sign_on": {boolean},
            "cos_override": {boolean},

            "quality_of_service": {
                "class_of_service": {
                    "explicit": "{string}",
                  | "map": {true},
                  | "preserve": {true}
                },

                "dscp": {
                    "explicit": {number},
                  | "map": {true},
                  | "preserve": {true}
                }
            }
        }
    }
}
```

Collection: Access Rules – IPv6

```
{
    "access_rules": [
        object#access_rule-ipv6-config,
        ...
    ]
}
```

Schema Attributes

Topics:

access_rule:

```
Type: object
Flags: -none-
Description: Access rule.
```

access_rules:

```
Type: array
Flags: -none-
Description: Access rule collection.
```

access_rule.ipv6:

```
Type: object
Flags: -none-
Description: IPv6 access rule.
```

access_rule.ipv6.from:

```
Type: string
Flags: key
Description: Zone object name.
```

access_rule.ipv6.to:

```
Type: string
Flags: key
Description: Zone object name.
```

access_rule.ipv6.action:

```
Type: string
Flags: key
Description: Set the action for this access rule.
```

access_rule.ipv6.source:

```
Type: object
Flags: key
Description: Source.
```

access_rule.ipv6.source.address:

```
Type: object
Flags: key
Description: Source address.
```

access_rule.ipv6.source.address.any:

```
Type: boolean (true)
Flags: key
Description: Any address.
```

access_rule.ipv6.source.address.name:

```
Type: string
Flags: key
Description: Address object name.
```

access_rule.ipv6.source.address.group:

```
Type: string
Flags: key
Description: Group address object name.
```

access_rule.ipv6.source.port:

```
Type: object
Flags: key
Description: Specify a source port for this Access Policy.
```

access_rule.ipv6.source.port.any:

```
Type: boolean (true)
Flags: key
Description: Any source service.
```

access_rule.ipv6.source.port.name:

```
Type: string
Flags: key
Description: Service object name.
```

access_rule.ipv6.source.port.group:

```
Type: string
Flags: key
Description: Service object group name.
```

access_rule.ipv6.service:

```
Type: object
Flags: key
Description: Specify a destination service for this Access Policy.
```

access_rule.ipv6.service.any:

```
Type: boolean (true)
Flags: key
Description: Any destination service.
```

access_rule.ipv6.service.name:

```
Type: string
Flags: key
Description: Service object name.
```

access_rule.ipv6.service.group:

```
Type: string
Flags: key
Description: Service object group name.
```

access_rule.ipv6.destination:

```
Type: object
Flags: key
Description: Destination.
```

access_rule.ipv6.destination.address:

```
Type: object
Flags: key
Description: Destination a destination address for this Access Policy.
```

access_rule.ipv6.destination.address.any:

```
Type: boolean (true)
Flags: key
Description: Any address.
```

access_rule.ipv6.destination.address.name:

```
Type: string
Flags: key
Description: Address object name.
```

access_rule.ipv6.destination.address.group:

```
Type: string
Flags: key
Description: Group address object name.
```

access_rule.ipv6.schedule:

```
Type: object
Flags: key
Description: Specify a schedule for this access policy.
```

access_rule.ipv6.schedule.always_on:

```
Type: boolean (true)
Flags: key
Description: Always on.
```

access_rule.ipv6.schedule.name:

```
Type: string
Flags: key
Description: Schedule object name.
```

access_rule.ipv6.users:

```
Type: object
Flags: key
Description: Specify users that are excluded from this access policy.
```

access_rule.ipv6.users.included:

```
Type: object
Flags: key
Description: Specify included users.
```

access_rule.ipv4.users.included.all:

```
Type: boolean (true)
Flags: key
Description: All users.
```

access_rule.ipv6.users.included.guests:

```
Type: boolean (true)
Flags: key
Description: Guest users.
```

access_rule.ipv6.users.included.administrator:

```
Type: boolean (true)
Flags: key
Description: Administrator.
```

access_rule.ipv6.users.included.name:

```
Type: string
Flags: key
Description: Local user object name.
```

access_rule.ipv6.users.included.group:

```
Type: string
Flags: key
Description: Local user group object name.
```

access_rule.ipv6.users.excluded:

```
Type: object
Flags: key
Description: Specify excluded users.
```

access_rule.ipv6.users.excluded.none:

```
Type: boolean (true)
Flags: key
Description: No users.
```

access_rule.ipv6.users.excluded.guests:

```
Type: boolean (true)
Flags: key
Description: Guest users.
```

access_rule.ipv6.users.excluded.administrator:

```
Type: boolean (true)
Flags: key
Description: Administrator.
```

access_rule.ipv6.users.excluded.name:

```
Type: string
Flags: key
Description: Local user object name.
```

access_rule.ipv6.users.excluded.group:

```
Type: string
Flags: key
Description: Local user group object name.
```

access_rule.ipv6.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

access_rule.ipv6.name:

```
Type: string
Flags: required
Description: Name.
```

access_rule.ipv6.comment:

```
Type: string
Flags: -none-
Description:
```

access_rule.ipv6.enable:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable this access rule.
```

access_rule.ipv6.reflexive:

```
Type: boolean (true|false)
Flags: -none-
Description: Configure a reflexive rule.
```

access_rule.ipv6.max_connections:

```
Type: number (uint8)
Flags: -none-
Description: Integer in the form: D OR 0xHH
```

access_rule.ipv6.logging:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable logging when this access rule is used.
```

access_rule.ipv6.management:

```
Type: boolean (true|false)
Flags: -none-
Description: Allow management traffic.
```

access_rule.ipv6.packet_monitoring:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable packet monitoring.
```

access_rule.ipv6.priority:

```
Type: object
Flags: -none-
Description: Set access rule priority
```

access_rule.ipv6.priority.auto:

```
Type: boolean (true)
Flags: -none-
Description: Set auto priority(priority = 0) for access rule.
```

access_rule.ipv6.priority.manual:

```
Type: number (uint32)
Flags: -none-
Description: Integer in the form: D OR 0xHHHHHHHH
```

access_rule.ipv6.tcp:

```
Type: object
Flags: -none-
Description: TCP.
```

access_rule.ipv6.tcp.timeout:

```
Type: number (uint32)
Flags: -none-
Description: Integer in the form: D OR 0xHHHHHHHH
```

access_rule.ipv6.udp:

```
Type: object
Flags: -none-
Description: UDP.
```

access_rule.ipv6.udp.timeout:

```
Type: number (uint32)
Flags: -none-
Description: Integer in the form: D OR 0xHHHHHHHH
```

access_rule.ipv6.fragments:

```
Type: boolean (true|false)
Flags: -none-
Description: Allow fragmented packets on this access rule.
```

access_rule.ipv6.botnet_filter:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable Botnet filter.
```

access_rule.ipv6.connection_limit:

```
Type: object
Flags: -none-
Description: Configure connection limit.
```

access_rule.ipv6.connection_limit.destination:

```
Type: object
Flags: -none-
Description: Enable connection limit for each destination IP address. Set to null or {} if
disabled/unconfigured.
```

access_rule.ipv6.connection_limit.destination.threshold:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

access_rule.ipv6.connection_limit.source:

```
Type: object
Flags: -none-
Description: Enable connection limit for each source IP address. Set to null or {} if
disabled/unconfigured.
```

access_rule.ipv6.connection_limit.source.threshold:

```
Type: number (uint16)
Flags: -none-
Description: Integer in the form: D OR 0xHHHH
```

access_rule.ipv6.flow_reporting:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable flow reporting.
```

access_rule.ipv6.geo_ip_filter:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable Geo-IP filter.
```

access_rule.ipv6.single_sign_on:

```
Type: boolean (true|false)
Flags: -none-
Description: Invoke single sign on to authenticate users.
```

access_rule.ipv6.cos_override:

```
Type: boolean (true|false)
Flags: -none-
Description: Allow 802.1p marking to override DSCP values.
```

access_rule.ipv6.quality_of_service:

```
Type: object
Flags: -none-
Description: Configure quality of service for rule.
```

access_rule.ipv6.quality_of_service.class_of_service:

```
Type: object
Flags: -none-
Description: Set 802.1p marking action. Set to null or {} if disabled/unconfigured.
```

access_rule.ipv6.quality_of_service.class_of_service.explicit:

```
Type: string
Flags: -none-
Description: Set explicit marking.
```

access_rule.ipv6.quality_of_service.class_of_service.map:

```
Type: boolean (true)
Flags: -none-
Description: Map marking.
```

access_rule.ipv6.quality_of_service.class_of_service.preserve:

```
Type: boolean (true)
Flags: -none-
Description: Preserve marking.
```

access_rule.ipv6.quality_of_service.dscp:

```
Type: object
Flags: -none-
Description: Set DSCP marking action.
```

access_rule.ipv6.quality_of_service.dscp.explicit:

```
Type: number (uint8)
Flags: -none-
Description: Integer in the form: D OR 0xHH
```

access_rule.ipv6.quality_of_service.dscp.map:

```
Type: boolean (true)
Flags: -none-
Description: Map marking.
```

access_rule.ipv6.quality_of_service.dscp.preserve:

```
Type: boolean (true)
Flags: -none-
Description: Preserve marking.
```

22

API: Route Policies – IPv4

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/route-policies/ipv4* **Schema:** *collection#route-policy-ipv4-config*	Empty	Required	Required	Required
URI: */api/sonicos/route-policies/ipv4/uuid/{UUID}* **Schema:** *collection#route-policy-ipv4-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: Route Policies – IPv4

```
{
    "route_policy": {
        "ipv4": {
            "interface": "{string}",
            "metric": {number},

            "source": {
                "any": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "destination": {
```

```
                "any": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "service": {
                "any": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "gateway": {
                "default": {true},
              | "name": "{string}",
              | "host": "{string}"
            },

            "uuid": "{string}",
            "name": "{string}",
            "disable_on_interface_down": {boolean},
            "vpn_precedence": {boolean},
            "auto_add_access_rules": {boolean},
            "probe": "{string}",
            "disable_when_probes_succeed": {boolean},
            "default_probe_state_up": {boolean},
            "comment": "{string}",
            "tcp_acceleration": {boolean},
            "wxa_group": "{string}"
        }
    }
}
```

Collection: Route Policies – IPv4

```
{
    "route_policies": [
        object#route-policy-ipv4-config,
        ...
    ]
}
```

Schema Attributes

Topics:

route_policy:

```
Type: object
Flags: -none-
Description: Route policy.
```

route_policies:

```
Type: array
Flags: -none-
Description: Route policy collection.
```

route_policy.ipv4:

```
Type: object
Flags: -none-
Description: IPv4 route policy.
```

route_policy.ipv4.interface:

```
Type: string
Flags: key
Description: Route interface name.
```

route_policy.ipv4.metric:

```
Type: number (uint8)
Flags: key
Description: Integer in the form: D OR 0xHH
```

route_policy.ipv4.source:

```
Type: object
Flags: key
Description: Set route policy source.
```

route_policy.ipv4.source.any:

```
Type: boolean (true)
Flags: key
Description: Any host.
```

route_policy.ipv4.source.name:

```
Type: string
Flags: key
Description: Host/network/range address object name.
```

route_policy.ipv4.source.group:

```
Type: string
Flags: key
Description: Group address object name.
```

route_policy.ipv4.destination:

```
Type: object
Flags: key
Description: Set route policy destination.
```

route_policy.ipv4.destination.any:

```
Type: boolean (true)
Flags: key
Description: Any host.
```

route_policy.ipv4.destination.name:

```
Type: string
Flags: key
Description: FQDN/host/network/range address object name.
```

route_policy.ipv4.destination.group:

```
Type: string
Flags: key
Description: Group address object name.
```

route_policy.ipv4.service:

```
Type: object
Flags: key
Description: Set route policy service.
```

route_policy.ipv4.service.any:

```
Type: boolean (true)
Flags: key
Description: Any service.
```

route_policy.ipv4.service.name:

```
Type: string
Flags: key
Description: Service object name.
```

route_policy.ipv4.service.group:

```
Type: string
Flags: key
Description: Service object group name.
```

route_policy.ipv4.gateway:

```
Type: object
Flags: key
Description: Set route policy gateway.
```

route_policy.ipv4.gateway.default:

```
Type: boolean (true)
Flags: key
Description: Default gateway 0.0.0.0
```

route_policy.ipv4.gateway.name:

```
Type: string
Flags: key
Description: Host address object name.
```

route_policy.ipv4.gateway.host:

```
Type: string (ip)
Flags: key
Description: IPv4 host address in the form: D.D.D.D
```

route_policy.ipv4.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

route_policy.ipv4.name:

```
Type: string
Flags: required
Description: Name.
```

route_policy.ipv4.disable_on_interface_down:

```
Type: boolean (true|false)
Flags: -none-
Description: Disable route when the interface is disconnected.
```

route_policy.ipv4.vpn_precedence:

```
Type: boolean (true|false)
Flags: -none-
Description: Allow VPN path to take precedence.
```

route_policy.ipv4.auto_add_access_rules:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable auto-add access rules.
```

route_policy.ipv4.probe:

```
Type: string
Flags: -none-
Description: Atom Object name.
```

route_policy.ipv4.disable_when_probes_succeed:

```
Type: boolean (true|false)
Flags: -none-
Description: Disable route when probe succeeds.
```

route_policy.ipv4.default_probe_state_up:

```
Type: boolean (true|false)
Flags: -none-
Description: Set probe default state to up.
```

route_policy.ipv4.comment:

```
Type: string
Flags: -none-
Description:
```

route_policy.ipv4.tcp_acceleration:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable permit TCP acceleration.
```

route_policy.ipv4.wxa_group:

```
Type: string
Flags: -none-
Description: WXA group name.
```

23

API: Route Policies – IPv6

Endpoint

Endpoint	HTTP method and body			
	GET	POST	PUT	DELETE
URI: */api/sonicos/route-policies/ipv6* **Schema:** *collection#route-policy-ipv6-config*	Empty	Required	Required	Required
URI: */api/sonicos/route-policies/ipv6/uuid/{UUID}* **Schema:** *collection#route-policy-ipv6-config*	Empty	—	Required	Ignored

Schema Structure

Topics:

Object: Route Policies – IPv6

```
{
    "route_policy": {
        "ipv6": {
            "interface": "{string}",
            "metric": {number},

            "source": {
                "any": {true},
              | "name": "{string}",
              | "group": "{string}"
            },

            "destination": {
```

```
            "any": {true},
          | "name": "{string}",
          | "group": "{string}"
        },

        "service": {
            "any": {true},
          | "name": "{string}",
          | "group": "{string}"
        },

        "gateway": {
            "default": {true},
          | "name": "{string}",
          | "host": "{string}"
        },

        "uuid": "{string}",
        "name": "{string}",
        "disable_on_interface_down": {boolean},
        "vpn_precedence": {boolean},
        "auto_add_access_rules": {boolean},
        "probe": "{string}",
        "disable_when_probes_succeed": {boolean},
        "default_probe_state_up": {boolean},
        "comment": "{string}"
    }
  }
}
```

Collection: Route Policies – IPv6

```
{
    "route_policies": [
        object#route-policy-ipv6-config,
        ...
    ]
}
```

Schema Attributes

Topics:

route_policy:

```
Type: object
Flags: -none-
Description: Route policy.
```

route_policies:

```
Type: array
Flags: -none-
Description: Route policy collection.
```

route_policy.ipv6:

```
Type: object
Flags: key
Description: IPv6 route policy.
```

route_policy.ipv6.interface:

```
Type: string
Flags: key
Description: Route interface name.
```

route_policy.ipv6.metric:

```
Type: number (uint8)
Flags: key
Description: Integer in the form: D OR 0xHH
```

route_policy.ipv6.source:

```
Type: object
Flags: key
Description: Set route policy source.
```

route_policy.ipv6.source.any:

```
Type: boolean (true)
Flags: key
Description: Any host.
```

route_policy.ipv6.source.name:

```
Type: string
Flags: key
Description: Host/network/range address object name.
```

route_policy.ipv6.source.group:

```
Type: string
Flags: key
Description: Group address object name.
```

route_policy.ipv6.destination:

```
Type: object
Flags: key
Description: Set route policy destination.
```

route_policy.ipv6.destination.any:

```
Type: boolean (true)
Flags: key
Description: Any host.
```

route_policy.ipv6.destination.name:

```
Type: string
Flags: key
Description: FQDN/host/network/range address object name.
```

route_policy.ipv6.destination.group:

```
Type: string
Flags: key
Description: Group address object name.
```

route_policy.ipv6.service:

```
Type: object
Flags: key
Description: Set route policy service.
```

route_policy.ipv6.service.any:

```
Type: boolean (true)
Flags: key
Description: Any service.
```

route_policy.ipv6.service.name:

```
Type: string
Flags: key
Description: Service object name.
```

route_policy.ipv6.service.group:

```
Type: string
Flags: key
Description: Service object group name.
```

route_policy.ipv6.gateway:

```
Type: object
Flags: key
Description: Set route policy gateway.
```

route_policy.ipv6.gateway.default:

```
Type: boolean (true)
Flags: key
Description: Default gateway 0.0.0.0/::
```

route_policy.ipv6.gateway.name:

```
Type: string
Flags: key
Description: Host address object name.
```

route_policy.ipv6.gateway.host:

```
Type: string (ip)
Flags: key
Description: IPv4 host address in the form: D.D.D.D IPv6 host address in the form:
HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH:HHHH.
```

route_policy.ipv6.uuid:

```
Type: string
Flags: key
Description: UUID in the form: HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

route_policy.ipv6.name:

```
Type: string
Flags: required
Description: Name.
```

route_policy.ipv6.disable_on_interface_down:

```
Type: boolean (true|false)
Flags: -none-
Description: Disable route when the interface is disconnected.
```

route_policy.ipv6.vpn_precedence:

```
Type: boolean (true|false)
Flags: -none-
Description: Allow VPN path to take precedence.
```

route_policy.ipv6.auto_add_access_rules:

```
Type: boolean (true|false)
Flags: -none-
Description: Enable auto-add access rules.
```

route_policy.ipv6.probe:

```
Type: string
Flags: -none-
Description: Atom Object name.
```

route_policy.ipv6.disable_when_probes_succeed:

```
Type: boolean (true|false)
Flags: -none-
Description: Disable route when probe succeeds.
```

route_policy.ipv6.default_probe_state_up:

```
Type: boolean (true|false)
Flags: -none-
Description: Set probe default state to up.
```

route_policy.ipv6.comment:

```
Type: string
Flags: -none-
Description:
```

24

SonicWall Support

Technical support is available to customers who have purchased SonicWall products with a valid maintenance contract and to customers who have trial versions.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. To access the Support Portal, go to https://www.sonicwall.com/support.

The Support Portal enables you to:

- View knowledge base articles and technical documentation
- View video tutorials
- Access MySonicWall
- Learn about SonicWall professional services
- Review SonicWall Support services and warranty information
- Register for training and certification
- Request technical support or customer service

To contact SonicWall Support, visit https://www.sonicwall.com/support/contact-support.

About This Document

Legend

WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

IMPORTANT, NOTE, TIP, MOBILE, or VIDEO: An information icon indicates supporting information.

SonicOS Reference
Updated - May 2019
Software Version - 6.5.4
232-004282-04 Rev A

Copyright © 2019 SonicWall Inc. All rights reserved.

SonicWall is a trademark or registered trademark of SonicWall Inc. and/or its affiliates in the U.S.A. and/or other countries. All other trademarks and registered trademarks are property of their respective owners

The information in this document is provided in connection with SonicWall Inc. and/or its affiliates' products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of SonicWall products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, SONICWALL AND/OR ITS AFFILIATES ASSUME NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON- INFRINGEMENT. IN NO EVENT SHALL SONICWALL AND/OR ITS AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF SONICWALL AND/OR ITS AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SonicWall and/or its affiliates make no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. SonicWall Inc. and/or its affiliates do not make any commitment to update the information contained in this document.

For more information, visit https://www.sonicwall.com/legal.

End User Product Agreement

To view the SonicWall End User Product Agreement, go to: https://www.sonicwall.com/en-us/legal/license-agreements.

Open Source Code

SonicWall is able to provide a machine-readable copy of open source code with restrictive licenses such as GPL, LGPL, AGPL when applicable per license requirements. To obtain a complete machine-readable copy, send your written requests, along with certified check or money order in the amount of USD 25.00 payable to "SonicWall Inc.", to:

General Public License Source Code Request
SonicWall Inc. Attn: Jennifer Anderson
1033 McCarthy Blvd
Milpitas, CA 95035