Web Security Service

Near Real-Time Log Sync Solution Brief

Revision: AUG.28.2020

Symantec
A Division of Broadcom

Copyrights

Near Real Time Access Log Sync

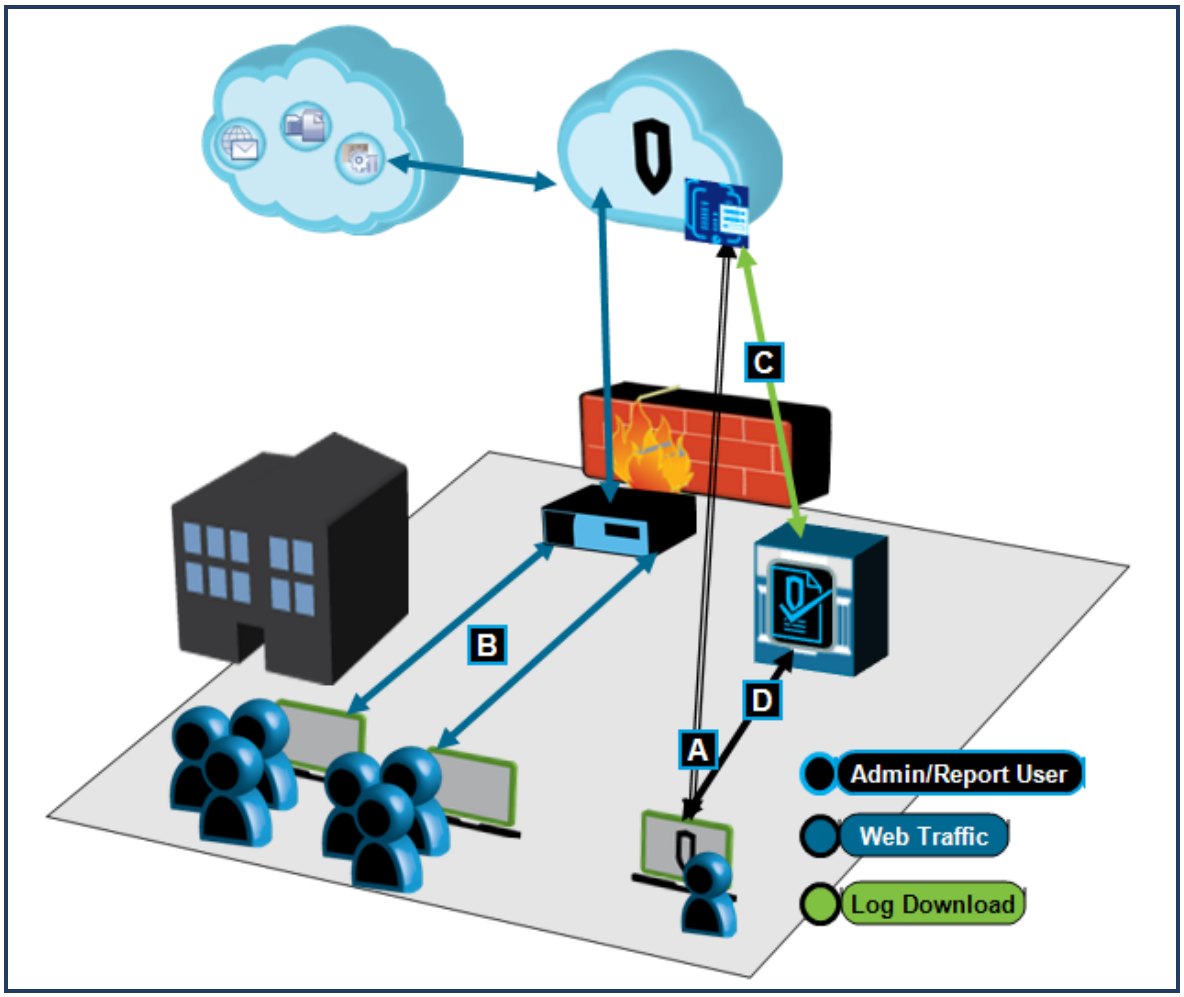
This document describes the Symantec WSS Sync API and how you can obtain near-real-time log data from the cloud service.

About Near-Real-Time Log Syncing

The Symantec Web Security Service offers two APIs (Download and Sync) for obtaining access log data from the cloud. The Download API restricts a client from receiving partial hour data, thus obtaining log data from the current hour is not possible with this API. The Sync API is an enhancement to the Download API. It allows a web client to obtain recently hardened log data from the cloud by downloading the current hour in smaller up-to-the-minute segments. You must provide the client utilities (scripts and programs) to facilitate both web download and subsequent log data processing. Then use Symantec Reporter or another third-party Security Information & Event Management (SIEM) engine to report on the log data.

Topography

The following is a brief description of the transaction. Subsequent sections provide greater detail.



A—IT Admin configures an on-premise client, with the Sync API and scripts, to communicate with the WSS. Valid WSS credentials are required.

B—Employees access web content and applications. The WSS compiles data into Access Logs.

C—The on-premise client requests a sequence of log lines from a (possibly) unbounded time range. The WSS sends a sequence of ZIP files.

D—IT/HR specialists are able to generate and view reports from near real-time data using Reporter or an SIEM engine.

About Access Log Delivery and Tokens

Log Delivery

The WSS delivers a ZIP archive that contains a collection of compressed text-based log lines (GZIPs).

The Download API only provides whole hour files, thus it uses a simpler file naming convention that specified only the year, month, day and hour of each GZIP hour file (the hour, minute, and second fields are still provided in the name, but comprised of zeros). The Sync API might download only a portion of any hour file in a given ZIP archive; therefore, in addition to these fields, its file naming convention also populates the hour, minute, and second fields of the file name. The Sync naming convention assures unique file names for any incremental hour download. Do not mix files obtained with both Download and Sync APIs as you might end up with apparently different filenames containing identical log data.

To prevent overload, any given sync download might limit the amount of data returned for a single request. For example, it is not necessary to stream previous weeks or months of data in a single response. The response status notifies the client if additional data is immediately available or if the entire request was satisfied. The WSS might also delay an aggressive client with a `429 too-many-requests` response code that informs the client not to send another request until waiting for an additional `retry-after` *`number of seconds`*. An overaggressive client is defined as one that is polling for any new data in the current hour more often than is reasonable. The WSS might also return a `service-unavailable` when an internal component is under maintenance.

The initial Sync API requires start and end times. The start date allows the client to start a Sync Stream at any archived hour. When the sync begins, the client uses the token returned in the previous response to receive contiguous log data. The client might also limit the range of any download by narrowing the start and end dates. The end date allows the client to stop a download at a given archive hour.

Tokens

The Sync API response introduces a token and status, which allows the client to resume a previous Sync session without data loss or duplication. The client receives the token and status following the compressed data in the successfully-downloaded zip file. The client then provides this token in the next Sync request to resume contiguous log downloads. If the client loses the token, it can no longer expect contiguous log data. The WSS does *not* maintain a record of the token content given to a client. The token is simply used like an API handle.

The WSS sends the ZIP archive file appended with the token and status.

- If the client receives all requested log lines, the status is done.
- If log lines are still available with the time range, the status is more. The client might repeat the request, which includes a new sync token.

Most ZIP archive decompression tools (such as Winzip) ignore the appended token and status at the end of the archive file.

The given start and end dates always override the token. The token helps the client obtain contiguous access log data and assists with the warning to the client about possible data loss. A token which falls within the given range is always honored, and a token which falls outside the given range, including expired dates, is always rejected.

About File Names

As previously discussed, the Sync API returns a set of compressed GZIP log hour files wrapped in a single ZIP archive file. Your web client must be able to name the ZIP archive file as it downloads the archive. To prevent name duplication, you might elect to use a date-based archive file name. For example:

```
cloud_archive_YYYYMMDDHHMMSS.zip
```

The Web Security Service names the GZIP hour files inside the archive. They always have date-based names. For example, a log file for hour 14 on January 23, 2017 is named:

```
cloud_54321_2017012314000001.log.gz
```

If that hour file was part of an incremental download, the next segment is named:

```
cloud_54321_2017012314000002.log.gz
```

And this naming convention continues.

Sync API Request Syntax

Example: The Sync API Request Syntax that you use on the local client to pull access logs from the WSS.

`https://portal.threatpulse.com/reportpod/logs/sync?startDate=`*start_date*`&endDate=`***end_date*`&token=`***token***

All three parameters are required, but dates can be `zero` and `token` can be `none`. The date is GMT.

> **Tip:** The GMT date details are discussed in the API examples section later in this document.

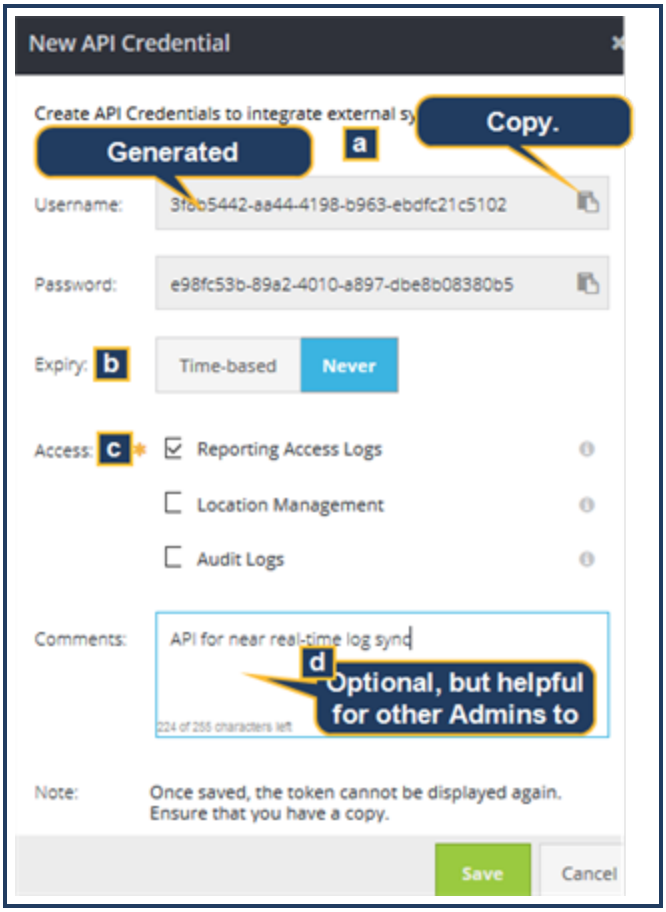
Next Step

- Proceed to "Create a User API Key" on page 9.

Create a User API Key

Create a User API Key that serves as the username and password in the Sync API syntax.

1. In **Service** mode, select **Account Maintenance > Integrations**.
2. Click **New Integration**.
3. Click **API Credentials**.
4. The WSS displays the New API Credential dialog, which contains the random characters **Username** and **Password**.



   a. Copy the **Username** and **Password** keys into a text file.
   b. Select the API **Expiry**.
      - **Time-based**—You define the date and time when this token expires.
      - **Never** expires.
   c. For the **Access** option, select **Reporting Access Logs**.
   d. Click **Save**.

Modify Web Client Application

To accept the access logs, the web client application must be able to validate the SyncAPI from the Web Security Service. Because the API URL does not contain access credentials, you must modify the web client application to use two HTTPS header fields.

How you add these headers depends on the Web/HTTPS client you use, such as curl or Wget. Furthermore, each client has its own syntax.

With your preferred application, add the following header fields. You must know the API name and password that you created in "Create a User API Key" on page 9, which you enter as the variables below.

- `X-APIUsername: api_name`
- `X-APIPassword: password`

After these are added, the HTTPS endpoint for the client's request, in this case the WSS portal, searches the HTTPS header for these headers and matches the API credentials. Upon a successful match, the client begins receiving the access logs.

Sync API Examples

A typical beginning Sync request to obtain all log data from a certain start date up to the present is similar to the following

```
https://portal.threatpulse.com/reportpod/logs/sync?&startDate= /
1522843200000&endDate=0&token=none
```

Date parameters can only be GMT on-the-hour boundaries in milliseconds-since-1970. All Symantec ProxySG appliance Access Log lines are GMT-based because of the global use of the product. If you live in an area where the time zones are not on GMT hour boundaries, you might need to adjust your date conversion algorithm accordingly. For example, your location is in the India Standard Time (IST), which differs from GMT by thirty minutes.

The `token=none` directive is required when there is no token. Start and end dates are also required, although they can be set to `0`. The client expects a `200` response code containing chunked data and an `X-sync-status: more` pair in the sync trailer if more data is available or `X-sync-status: done` if all available data was returned. For sequential downloads, the client obtains the token from the `X-sync-token` value from the sync trailer to use in the next request.

The next Sync request might look similar to this

```
https://portal.threatpulse.com/reportpod/logs/sync? /
&startDate=1522843200000&endDate=0&token=4E0329AEB104A3625A6f7D26C41E735F
```

While the `X-sync-status` continues to be `more`, the client might continue to request more data, updating the token each time with the value returned in the `X-sync-token`. When `X-sync-status` eventually changes to `done`, the client pauses for a reasonable amount of time before sending the next poll request. The WSS might send `429` error codes to delay an overaggressive client until a reasonable amount of time has elapsed.

The client cannot not discard its current token until a new token is obtained. For contiguous data downloads, the client application might store its token so that the application can obtain it after a reboot.

About Response Codes

This section provides response code implications for Sync API transactions with the Symantec Web Security Service.

Request Codes

The client must be informed about various failures and possible recovery options. The Sync Token is associated with a specific set of hardened data. If that set ever becomes unavailable because of eventual data expiration or unexpected catastrophic failure, the token is no longer valid. Because Symantec expects neither of these cases to occur during normal usage, client recovery is kept intentionally simple and therefore might not entirely prevent duplicated or lost data. In both of these cases, the WSS returns error 410 (Gone) to inform the client that the token is no longer valid and should be thrown away. The client might obtain a new token by sending a request with a new start date and no token. If the client had already obtained partial data for the start date, that data is duplicated in the new download.

The association between dates and tokens create a number of possible conditions. The client might or might not provide a start date, end date, or token. And the token might fall before, within, or after the start or end dates. The following table provides the possible client request conditions and subsequent WSS actions.

Start and End	No Token	Token Before	Token Within	Token After
0,0	BAA,SAA	NA	RAT,SAA	NA
S,0	BAS,SAA	410	RAT,SAA	NA
0,E	BAA,SAE	NA	RAT,SAE	410
S,E	BAS,SAE	410	RAT,SAE	410

- BAS—Return 200 and begin at start date
- RAT—Return 200 and resume at token
- BAA—Return 200 and begin at archive head
- SAA—Stop at archive tail
- SAE—Stop at end date
- NA—Not Applicable
- 410—Return error code with no data

The WSS streams the log download using HTTP chunked transfer encoding. The initial response might be a 400- or 500-level error if the request is invalid or cannot be processed immediately. Otherwise, the WSS replies with a 200-level response and begins streaming the download archive. The ZIP archive builds *on the fly*. In addition to the `Sync Token`, the `Sync Trailer` also contains a `Sync Status`. If some portion of building or streaming the archive fails, the WSS attempts to note the failure in that status. Thus, the initial 200-level response in the first packet does not guarantee that the download produces a complete ZIP archive. The client must always check the final status in the trailer to know if the download completed successfully and if the chunked archive is expected to unpack correctly.

All 500-level responses suggest that the client can repeat the last token when the service becomes available. 400-level responses require more scrutiny. A 410 (Gone) response means the token is no longer valid and should be discarded. The 429

(Too Many Requests) response means the WSS is unwilling to service the client at the moment. The WSS might throttle the client during maintenance, when failures occur, or when the client is polling too often within the current hour. The HTTP header provides a `Retry-After` field to indicate how many seconds the client pauses until sending the next request. The default throttle is expected to be around five (5) minutes. Symantec/Symantec recommends that customers who create multiple copies of their cloud service archive data use a single download client and multiplex the data after it is downloaded. Thus, the WSS imposes the throttle across all clients of the same customer regardless of client endpoint or API Key.

The Sync Trailer and HTTP Codes

The successful Sync API response is a compressed archive file with a small sync trailer appended to it. The sync trailer does not impact the decompression of the archive file, but contains two key-value pairs: `X-sync-status` and `X-sync-token`. The Sync Trailer is only sent in the final chunk encoding, and chunks are only returned when the initial response is `200`. The client examines the token and status to discover the final status of the downloaded archive and to obtain the next token.

The `X-sync-token` value is the token string that is used in the next request. It might match the previous token if no new data was available or if an internal failure occurred causing the archive to abort prematurely.

The `X-sync-status` value is `done`, `more`, or `abort`.

- `done`—The request was satisfied and no more data is currently available within the specified date range. “
- `more`—The archive was intentionally limited in size and another request with the new token would obtain more contiguous data immediately.
- `abort`—The WSS experienced a failure while building the archive. The archive might or might not contain usable data, although an aborted archive might still produce a number of valid log files before the archive fails to unpack completely. The final log file in any aborted archive is always discarded because of unknown missing data. A subsequent Sync request could drop the previous token and simply provide a new start date where the aborted archive left off. Otherwise, the entire archive could be discarded and the previous token could be used for a subsequent retry of the entire download.

Different HTTP codes and Sync Trailer values allow the client to efficiently use the Sync API.

HTTP Codes	Description

`200 OK`
`X-sync-status: more`
`X-sync-token:` *`new_token`*
	Data was returned and more data is available. The WSS chose to limit the amount of data returned in this response. The client knows that another request would obtain more data immediately. The client provides the new token in the next request.

`200 OK`
`X-sync-status: done`
`X-sync-token:` *`new_token`*
	Data was returned and no more data is available. If an end date was given, the download is complete. If no end date was given, the client pauses before polling for new data. The client provides the new token in the next request.

`200 OK`
`X-sync-status: done`
`X-sync-token:` *`same_token`*
	No data was returned and no more data is available. If a date range was given, the WSS found no data within the range. If no end date was given, the client pauses before polling for new data. If polling, the client reuses the same token until a new token is provided.

HTTP Codes	Description

`200 OK`
`X-sync-status: abort`
`X-sync-token: *same_token*`
	Data was returned and more data is available after the WSS issue is resolved. The service experienced an internal issue while building the ZIP archive. Much of the download might be usable, except for the last file in the archive. The client might begin a new session by requesting a new start date without a token. Otherwise, the client reuses the same dates and token until the download is successful.

`400 Bad Request`	The start date is later than the end date; bad syntax; or some other request error. The client verifies it is properly using the Sync API.
`410 Gone`	The token is no longer within the given dates or the token references expired data and the token's associated WSS data is no longer available. The token is no longer valid and is discarded. The client might obtain a new token by requesting a new start date without a token.
`429 Too Many Requests`	The WSS is unwilling to service the client until a reasonable time has elapsed. The `Retry-After` field shows the remaining wait period in seconds. The default is expected to be around five (5) minutes.
`500 Internal Error`	An internal WSS error prevented the download. The client might need to wait a while before repeating the request.
`503 Service Unavailable`	No resources are currently available to service the request; or a WSS component is temporarily offline. The service might restrict the total number of downloads currently in progress. The client waits before repeating the request.