Tizen Web Protection

API Specification

Document version 1.0.5

Document Information

Document Details

Revision	1.0.4
Author	McAfee Mobile Team

Revision Information

Revision	Revision Date	Author	Details
1.0.0	09/26/2012	McAfee Mobile Team	Created
1.0.1	10/21/2021	McAfee Mobile Team	Change architecture block diagram.
1.0.2	11/27/2012	McAfee Mobile Team	Add library handle for multiple instance need.
1.0.3	01/26/2013	McAfee Mobile Team	Add license
1.0.4	02/13/2013	McAfee Mobile Team	Flag configuration key/id parameters as optional.
1.0.5	1/6/2014	McAfee Mobile Team	Reformat WP tables

Contents

Terms, Abbreviations, Definitions, Conventions

Items	Description
SDK	Software Development Kit
API	Application Programming Interface
Client ID	A unique application ID generated and assigned by security vendor to the SDK integrator and used to initialize the SDK.
Client Key	A unique client key generated and assigned by security vendor to the SDK integrator and used to initialize the SDK.
Application	Executable provided by either system or third-party
Rating Score or URL Score	The rating score is an integer number assigned by security vendor that reflects a website's reputation. The higher the absolute value of a score, the more dangerous a website is. Developers should always use absolute score values to determine website reputation. Negative values correspond to phishing sites. In the case of a negative score, a website's reputation should still be determined using the absolute value of the score. In general, the absolute values of the scores can be divided into four categories: 'High Risk' (or 'Red'), 'Medium Risk' (or 'Yellow'), 'Low Risk' (or 'Green'), and 'Unverified' (or 'Grey'). **Note:** 'Unverified' means that a rated website was analyzed but a score representing the risk of visiting that website cannot be clearly assigned.
DLA	Deep link analysis. Web Protection can detect URL redirection chains and report the URL that ends the redirection chain back to the client. This feature is useful when such chains lead from websites with good reputations to dangerous or exploited webpages. When the SDK is used to look up the URL at the start of a chain, the back end will report the ending URL and the score of the ending URL.
URL Lookup	A process of determining a URL's reputation using the Web Protection SDK.

Overview

This document is a guide to developing applications that use Web Protection Services. It describes:

- The minimum requirements for developing applications with the Web Protection Services SDK
- Instructions on how to use the SDK with applications and interpret the results
- A complete list of the API elements

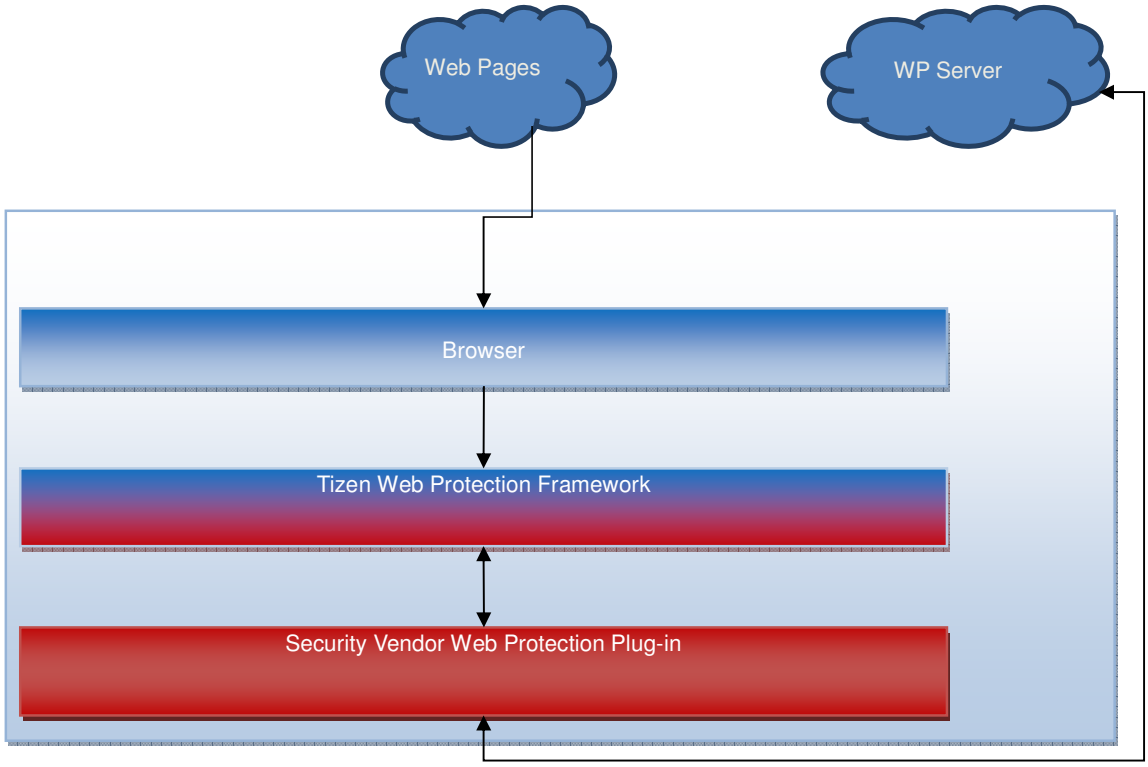


Figure 1 Overview of Web Protection Service SDK

Web Protection Service can protect users around the world from web-based malware threats, browser exploits, and identity theft.

With the spread of connected devices and the popularity of related apps, mobile vendors and businesses are looking for new ways to use the 'cloud' in order to provide seamless and fully integrated web protection. This document has responded to these trends by introducing Web Protection Services. Partners and businesses can now protect their users by directly scanning all external, website links against Web Protection Services and assessing their safety.

Web Protection Services SDK is flexible and can be adapted to a variety of different protection goals. For example:

- A website owner may want to integrate Web Protection Services to scan external links for up-to-the-minute warnings about target destinations when users click on them.

- An application developer may want to integrate Web Protection Services to scan attachments or links for known malicious or suspicious risks.

Integrators using Web Protection Services can configure what users will experience if a link or attachment is found to be risky. Integrators can add notes in line with links and attachments to illustrate the risk of clicking on links. For convenience, Web Protection Services also provides partners a redirection page for risky links, but partners have complete control over how to handle a risky site.

Web Protection Services give direction about the reputation and category of any given URL. This provides site analysis protection to users without them having to install security software. The Web Protection Services SDK lets you use Web Reputation cloud to determine:

- The reputation of one or more URLs
- The category of one or more URLs
- If URLs should be blocked, providing a URL to redirect users to a block page

The Web Protection Services SDK uses 'cloud' technologies to provide URL reputation information. For the Web Protection Services SDK to work properly, the licensee's applications must have Internet access.

SDK Concepts

External Dependencies

The Web Protection Services SDK was designed to minimize dependencies on external libraries and APIs. The SDK itself does not explicitly require any additional libraries and does not directly invoke any platform dependent or standard C APIs. Instead, the SDK requires applications to pass a few pointers to functions to implement the following functionality:

- Memory allocation
- Memory de-allocation
- Random number generation

The first two functions are used by the SDK to allocate and free temporary resources. The last function is used to obfuscate the outgoing HTTP requests.

Handles

The SDK uses handles to refer to internal objects that implement a specific subset of functionality. For instance, `TWPConfigurationHandle` represents a configuration object that maintains a set of variables internally, which allow for altering the behavior of other functions and objects.

Policies

Policies allow for validating a predefined set of website categories against URL rating information obtained after a URL lookup. For example, a policy can be defined to detect 'Nudity' and 'Pornography' categories. After each successful URL lookup, this policy can be compared to (or 'validated' against) a specific URL rating. If the rating corresponds to a website (URL) serving the content that belongs to one of the above categories, the policy is considered 'violated' and the SDK will indicate this to the caller. For more details, refer to API description.

Adding Web Protection Services SDK to your application

The SDK is distributed as a single, static object library file accompanied by one include file. To integrate the Web Protection Services SDK into your application, include the `TWP.h` in one of your source files and link it with the library.

Using Web Protection Services APIs

Examples

Start off with one of the example application provided with the SDK. Each example can be compiled using the provided makefiles or IDE project files. To successfully execute a compiled example application, you will have to specify a Client ID and Client Key pair in the example source file before compiling in the `client_id` and `client_key` fields of the `TWPConfiguration` structure.

1. Initializing SDK

Initialize an instance of `TWPAPIInit` structure and pass it to `TWPInitLibrary`:

```
TWPAPIInit tInit;
TWPLIB_HANDLE hLIb;
tInit.api_version = TWP_API_VERSION;
tInit.memallocfunc = malloc;
tInit.memfreefunc = free;
hLib = TWPInitLibrary( &tInit );
```

2. Creating a configuration handle

Initialize the `TWPConfiguration` structure using the Client ID and Client Key provided by security vendor, and pass it to `TWPConfigurationCreate`:

```
TWPConfigurationHandle hCfg;
TWPConfiguration tCfg;
tCfg.config_version = TWP_CONFIG_VERSION;
tCfg.client_id = "<your ID provided by security vendor>";
tCfg.client_key = "< Key provided by security vendor>";
tCfg.obfuscate_request = 1;
tCfg.randomfunc = random;
...
TWPConfigurationCreate( hLib, &tCfg, &hCfg );
```

You can have multiple configuration classes for each clientID and clientKey pair that you use.

3. Implementing HTTP functionality

The SDK requires applications to implement all functionality related to sending and receiving data over HTTP. Four callback functions have to be implemented by the application. Function pointers are passed to the SDK using the fields of `TWPRequest` structure. A pointer to the original structure is passed to every callback function as a parameter.

4. Looking up URLs

After initializing the `TWPRequest` structure, invoke `TWPLookupUrls`.

Note: All strings representing the URLs passed to this call have to be properly normalized and pynicoded, if required. If strings passed to this function contain non-7-bit ASCII characters, the result of the lookup is undefined.

5. Uninitialization

For the uninitialization process, all handles obtained during the previous stages must be destroyed using corresponding APIs. Finally, uninitialize the library by calling `TWPUninitLibrary`.

Library

The delivery of Tizen Web Protection API should be a .so library: `libtwp.so`

Initialize Functions

Summary

Methods	
`TWPLIB_HANDLE`	`TWPInitLibrary(TWPAPIInit *pApiInit)`
	Initialize Tizen Web Protection library.
`void`	`TWPUninitLibrary(TWPLIB_HANDLE hLib)`
	Uninitializes the library and cleans up all resources.
`TWP_RESULT`	`TWPConfiguratinCreate(TWPLIB_HANDLE hLib, TWPConfiguration *pConfigure, TWPConfigurationHandle *phConfigure)`
	Create and initializes a Web Protection Service configuration handle.
`TWP_RESULT`	`TWPConfigurationDestroy(TWPLIB_HANDLE hLib, TWPConfigurationHandle *phConfigure)`
	Destroys and reclaims all memory from a configuration handle.

Methods

```
TWPLIB_HANDLE TWPInitLibrary (TWPAPIInit *pApiInit)
```

Initializes the Web Protection Services SDK library.

This function must be called once, usually during application startup, and before any other function in this library is used. This function cannot be called more than once.

Parameters

`[in] pApiInit`	Pointer to the initialized `TWPAPIInit` structure

Returns

Valid library handle	Success
`INVALID_TWPLIB_HANDLE`	Failed to initialize library.

```
void TWPUninitLibrary (TWPLIB_HANDLE hLib)
```

Uninitializes the library and cleans up all resources.

This function must be called only once and is usually called during application shutdown. After this call, no other Web Protection Services SDK function in this library can be used.

Parameters

`[in] hLib`	Library handle

Returns

None.

```
TWP_RESULT TWPConfiguratinCreate(TWPLIB_HANDLE hLib,
                                 TWPConfiguration *pConfigure,
                                 TWPConfigurationHandle *phConfigure)
```

Creates and initializes a Web Protection Services configuration handle.

`TWPConfiguration` handle represents the configuration object that must be passed to other Web Protection Services SDK API functions. It is possible to have multiple configuration handles in one application. To release any memory allocated by this function, `TWPConfigurationDestroy` must be called. This function can only be called one time for each handle.

Parameters

`[in] hLib`	Library handle
`[in] pConfigure`	properly initialized `TWPConfiguration` structure.
`[out] phConfigure`	pointer to a configuration handle. When the function completes successfully, the memory pointed by `phConfigure` contains the value of the newly created configuration handle.

Returns

`TWP_INVALID_PARAMETER`	if any of the following errors occur: • `pConfigure` is `NULL`. • `phConfigure` is `NULL`. • `TWPConfiguration.randomfunc` is `NULL` and `TWPConfiguration.obfuscate_request` is not `NULL`.
`TWP_INVALID_VERSION`	`TWPConfiguration.config_version` is not `TWP_CONFIG_VERSION`.
`TWP_NOMEM`	The function failed to allocate memory required to complete initialization of the configuration object.
`TWP_SUCCESS`	The configuration object was successfully created and initialized.

```
TWP_RESULT TWPConfigurationDestroy(TWPLIB_HANDLE hLib,
                                   TWPConfigurationHandle *phConfigure)
```

Destroys and reclaims all memory from a configuration handle.

This function reclaims the memory used by a configuration handle. The handle must have been previously initialized with `TWPConfigurationCreate`. This function can be called one time for each initialized handle.

Parameters

`[in] hLib`	Library handle
`[in] phConfigure`	Pointer to a configuration handle. When the function completes successfully, the memory pointed

to by `phConfigure` is set to `NULL`.

Returns

`TWP_INVALID_HANDLE`	The value pointed to by `phConfigure` does not represent a valid configuration handle.
`TWP_SUCCESS`	The handle was successfully destroyed.

Policy Functions

Summary

Methods	
`TWP_RESULT`	`TWPPolicyCreate(TWPLIB_HANDLE hLib, TWPCategories *pCategories, unsigned int uCount, TWPPolicyHandle *phPolicy)`
	Creates and initializes a policy handle.
`TWP_RESULT`	`TWPPolicyValidate(TWPLIB_HANDLE hLib, TWPPolicyHandle hPolicy, TWPURLRatingHandle hRating, int *piViolated)`
	Compares the categories assigned by caller to the URL represented by `handle_rating` with the categories assigned to the policy handle.
`TWP_RESULT`	`TWPPolicyGetViolations(TWPLIB_HANDLE hLib, TWPPolicyHandle hPolicy, TWPURLRatingHandle hRating, TWPCategories **ppViolated, unsigned int *piLength)`
	Retrieves all categories common to the policy and URL rating.
`TWP_RESULT`	`TWPPolicyDestroy(TWPLIB_HANDLE hLib, TWPPolicyHandle *phPolicy)`
	Destroys and reclaims all memory from a policy handle.

Methods

```
TWP_RESULT TWPPolicyCreate (TWPLIB_HANDLE hLib,
                            TWPCategories *pCategories,
                            unsigned int uCount,
                            TWPPolicyHandle *phPolicy)
```

Creates and initializes a policy handle.

Parameters

`[in] hLib`	Library handle.
`[in] pCategories`	An array of TWPCategories values.
`[in] uCount`	Length of the categories array.
`[out] phPolicy`	A pointer to the location where the initialized policy handle value should be stored.

Returns

`TWP_INVALID_PARAMETER`	if one of the following errors occur:

- pCategories in NULL.
- uCount is 0.
- phPolicy is NULL.

TWP_NOMEM	if the function fails to allocate memory for the policy object.
TWP_SUCCESS	if the function is completed successfully and handle_policy is initialized.

```
TWP_RESULT TWPPolicyValidate(TWPLIB_HANDLE hLib,
                             TWPPolicyHandle hPolicy,
                             TWPURLRatingHandle hRating,
                             int *piViolated)
```

Compares the categories assigned by security vendor to the URL represented by hRating with the categories assigned to the policy handle.

If category intersection exists, the piViolated parameter is set to a non-zero value.

Parameters

[in] hLib	Library handle.
[in] hPolicy	A policy handle obtained from TWPPolicyCreate.
[in] hRating	A URL rating handle obtained using TWPResponseGetUrlRatingByIndex or TWPResponseGetUrlRatingByUrl.
[out] piViolated	Set to a non-zero value if there is an intersection between the policy and URL rating categories.

Returns

TWP_INVALID_HANDLE	if hPolicy or hRating does not correspond to a valid handle.
TWP_INVALID_PARAMETER	piViolated is NULL.
TWP_SUCCESS	Function completed successfully.

```
TWP_RESULT TWPPolicyGetViolations(TWPLIB_HANDLE hLib,
                                  TWPPolicyHandle hPolicy,
                                  TWPURLRatingHandle hRating,
                                  TWPCategories **ppViolated,
                                  unsigned int *piLength)
```

Retrieves all categories common to the policy and URL rating.

Effectively, this function retrieves category intersections between the policy and URL rating.

Parameters

[in] hLib	Library handle.
[in] hPolicy	A policy handle obtained from TWPPolicyCreate.
[in] hRating	A URL rating handle obtained using TWPResponseGetUrlRatingByIndex or TWPResponseGetUrlRatingByUrl.
[out] ppViolated	An array of all common categories. This array is allocated using TWPAPIInit::memallocfunc and has to be deallocated by the caller.
[out] piLength	The length of ppViolated.

Returns

TWP_INVALID_HANDLE	if hPolicy or hRating does not correspond to a valid handle.
TWP_INVALID_PARAMETER	ppViolated or piLength is NULL.
TWP_NOMEM	ppViolated cannot be allocated.
TWP_SUCCESS	Function completed successfully.

```
TWP_RESULT TWPPolicyDestroy(TWPLIB_HANDLE hLib,
                            TWPPolicyHandle *phPolicy)
```

Destroys and reclaims all memory from a policy handle.

This function reclaims the memory used by a policy handle. The handle must have been previously initialized with TWPPolicyCreate. This function can be called one time for each initialized handle.

Parameters

[in] hLib	Library handle.
[in, out] phPolicy	A pointer to a policy handle. When the function completes successfully, the memory pointed to by phPolicy is set to NULL.

Returns

TWP_INVALID_HANDLE	hPolicy does not correspond to a valid handle.
TWP_SUCCESS	Function completed successfully.

Lookup Functions

Summary

Methods	
TWP_RESULT	TWPLookupUrls(TWPLIB_HANDLE hLib, TWPConfigurationHandle hConfigure, TWPRequest *pRequest, int iRedirUrlFlag, const char **ppUrls, unsigned int uCount, TWPResponseHandle *phResponse)
	Submits the provided URLs to the server for rating.
TWP_RESULT	TWPResponseWrite(TWPLIB_HANDLE hLib, TWPResponseHandle hResponse, const void *pData, unsigned uLength)
	Callers implementing asynchronous requests must use this function to pass chunks of server response as they become available.
TWP_RESULT	TWPResponseGetUrlRatingByIndex(TWPLIB_HANDLE hLib, TWPResponseHandle hResponse, unsigned int uIndex, TWPUrlRatingHandle *phRating)
	Retrieves a URL rating object handle from the response object using the URL index as it appeared in the call to TWPLookupUrls
TWP_RESULT	TWPResponseGetUrlRatingByUrl(TWPLIB_HANDLE hLib,

	TWPResponseHandle hResponse, const char *pUrl, unsigned int uUrlLength, TWPUrlRatingHandle *phRating)
	Retrieves a URL rating object handle from the response object using the full URL string as it appeared in the call to TWPLookupUrls
TWP_RESULT	TWPResponseGetRedirUrlFor(TWPLIB_HANDLE hLib, TWPResponseHandle hResponse, TWPUrlRatingHandle hRating, TWPPolicyHandle hPolicy, char **ppUrl, unsigned int *piLength)
	This function can be used by the application to obtain a URL for the security vendor blocking page.
TWP_RESULT	TWPResponseGetUrlRatingsCount(TWPLIB_HANDLE hLib, TWPResponseHandle hResponse, unsigned int *puCount)
	Returns the number of URL ratings currently available in the response object.
TWP_RESULT	TWPResponseDestroy(TWPLIB_HANDLE hLib, TWPResponseHandle *phResponse)
	Destroys and reclaims all memory from a response handle.

Methods

```
TWP_RESULT TWPLookupUrls(TWPLIB_HANDLE hLib,
                         TWPConfigurationHandle hConfigure,
                         TWPRequest *pRequest,
                         int iRedirUrlFlag,
                         const char **ppUrls,
                         unsigned int uCount,
                         TWPResponseHandle *phResponse)
```

Submits the provided URLs to the server for rating.

This function uses TWPRequest structure for invoking an application specific implementation of HTTP channel. Application developers may choose between two different implementation models of HTTP communication: synchronous and asynchronous. The function checks TWPRequest.receivefunc to determine the desired model; NULL value specifies the asynchronous model.

Synchronous mode.

In the synchronous operation mode, the function invokes TWPRequest::sendfunc and TWPRequest::receivefunc, one right after the other, expecting the entire HTTP transaction to be completed between the calls. Upon successful completion, the phResponse will point to a valid response handle that can be used to analyze results.

Asynchronous mode.

In the asynchronous mode, the function invokes TWPRequest::sendfunc and returns immediately with TWP_SUCCESS. Upon completion, phResponse is NULL. The application is supposed to complete the HTTP transaction while calling TWPResponseWrite as response data becomes available. When all data was read, TWPResponseWrite must be called again with zero data length to signal the end

transaction.

Parameters

[in] hLib	Library handle.
[in] hConfigure	Configuration handle previously created using TWPConfigurationCreate.
[in] pRequest	Pointer to initialized TWPRequest structure.
[in] iRedirUrlFlag	non zero value will instruct the server to provide a landing page URL to which blocked URLs can be redirected.
[in] ppUrls	An array of 7 bit ASCII character strings representing URLs to obtain the rating for. **Note:** All URLs have to be normalized before submission (see RFC 3986) and pynicoded if required.
[in] uCount	Length of the ppUrls array.
[out] phResponse	For synchronous requests, a pointer to the location where the response object handle will be stored upon completion. It can be NULL for asynchronous requests.

Returns

TWP_INVALID_HANDLE	hConfigure is not a valid configuration handle.
TWP_INVALID_VERSION	TWPRequest.request_version is not TWP_REQUEST_VERSION
TWP_INVALID_PARAMETER	One of the following errors occur: • ppUrls is NULL. • any string in the ppUrls array is NULL or has zero length. • uCount is 0. • pRequest is NULL. • TWPRequest.receivefunc is not NULL and phResponse is NULL (if TWPRequest.receivefunc is NULL, hResponse can be NULL).
TWP_NOMEM	Function failed to allocate memory.
TWP_SUCCESS	Function completed successfully.
other	Any integer value other than TWP_SUCCESS returned by TWPRequest::setmethodfunc, TWPRequest::seturlfunc and TWPRequest::sendfunc functions.

```
TWP_RESULT TWPResponseWrite(TWPLIB_HANDLE hLib,
                            TWPResponseHandle hResponse,
                            const void *pData,
                            unsigned uLength)
```

Callers implementing asynchronous requests must use this function to pass chunks of server

response as they become available.

When the entire response is read, this function must be called again with zero `uLength` to signal end of data. This function can only be called after a successful call to `TWPLookupUrls`.

Parameters

`[in] hLib`	Library handle.
`[in] hResponse`	Response handle that was passed by `TWPLookupUrls` to `TWPRequest::sendfunc`.
`[in] pData`	Pointer to response data buffer.
`[in] uLength`	Size in bytes of the buffer pointed by `pData`.

Returns

`TWP_INVALID_HANDLE`	`hResponse` does not correspond to a valid handle.
`TWP_ERROR`	Internal error occurred.
`TWP_INVALID_RESPONSE`	Server response is invalid. This can usually happen when the entire response was not passed to the SDK.
`TWP_SUCCESS`	Function successfully processed the server reply.

```
TWP_RESULT TWPResponseGetUrlRatingByIndex(TWPLIB_HANDLE hLib,
                                          TWPResponseHandle hResponse,
                                          unsigned int uIndex,
                                          TWPUrlRatingHandle *phRating)
```

Retrieves a URL rating object handle from the response object using the full URL string as it appeared in the call to `TWPLookupUrls`.

After the server reply was processed, the response object represented by `hResponse` contains information about all URLs submitted to the server in `TWPLookupUrls`. This function allows access to information specific to `uIndex` using the URL rating object. This function can be called only if `TWPLookupUrls` and `TWPResponseWrite` are completed successfully. This method is slightly faster than `TWPResponseGetUrlRatingByUrl`.

Note:

The URL rating object referenced by `phRating` does not need to be released; it is valid and will not change until the parent response object is destroyed using `TWPResponseDestroy`.

Parameters

`[in] hLib`	Library handle.
`[in] hResponse`	Response handle that obtained in the call to `TWPLookupUrls`.
`[in] uIndex`	The index of URL in the `ppUrls` in call to `TWPLookupUrls`.
`[out] phRating`	Pointer to the memory location where to store the handle if the function completes successfully.

Returns

`TWP_INVALID_PARAMETER`	`phRating` is `NULL`.
`TWP_INVALID_HANDLE`	`hResponse` does not represent a valid response object.
`TWP_NO_DATA`	Index is out of bound.
`TWP_SUCCESS`	Function successfully and `phRating` is initialized.

```
TWP_RESULT TWPResponseGetUrlRatingByUrl(TWPLIB_HANDLE hLib,
                                        TWPResponseHandle hResponse,
                                        const char *pUrl,
                                        unsigned int uUrlLength
                                        TWPUrlRatingHandle *phRating)
```

Retrieves a URL rating object handle from the response object using a full URL string as it appeared in the call to `TWPLookupUrls`.

After the server reply is processed, the response object represented by `hResponse` contains information about all URLs submitted to the server in `TWPLookupUrls`. This function gives access to information specific to URL using the URL rating object. This function can be called only if `TWPLookupUrls` and `TWPResponseWrite` (in case of asynchronous request) are completed successfully. This method is slightly slower than `TWPResponseGetUrlRatingByIndex`.

Note:

The URL rating object referenced by `phRating` does not need to be released; it is valid and will not change until the parent response object is destroyed using `TWPResponseDestroy`.

Parameters

`[in] hLib`	Library handle.
`[in] hResponse`	Response handle that obtained in the call to `TWPLookupUrls`.
`[in] pUrl`	URL string as it appeared in the `urls` array in call to `TWPLookupUrls`.
`[in] uUrlLength`	The length of the `URL` parameter. If the length is unknown and the string is null terminated, specify zero.
`[out] phRating`	Pointer to the memory location where to store the handle if the function completes successfully.

Returns

`TWP_INVALID_PARAMETER`	`phRating` is `NULL`.
`TWP_INVALID_HANDLE`	`hResponse` does not represent a valid response object.
`TWP_NO_DATA`	`url` does not correspond to an existing URL rating object.
`TWP_SUCCESS`	Function successfully and `phRating` is initialized.

```
TWP_RESULT TWPResponseGetRedirUrlFor(TWPLIB_HANDLE hLib,
                                     TWPResponseHandle hResponse,
                                     TWPUrlRatingHandle hRating,
                                     TWPPolicyHandle hPolicy,
                                     char **ppUrl,
                                     unsigned int *piLength)
```

This function can be used by the application to obtain a URL for the security vendor blocking page.

Blocking pages can be used by applications that want to block users from navigating to a URL that violates one of the defined policies. The returned string must be deallocated by the

application using TWPFnMemFree function.

Parameters

[in] hLib	Library handle.
[in] hResponse	Response object handle.
[in] hRating	URL rating object handle.
[in] hPolicy	Policy handle.
[out] ppUrl	Pointer to a null terminated string representing the redirection URL.
[out] piLength	(optional) A pointer to a variable that receives the length of the URL. If the length is not required, set to NULL.

Returns

TWP_INVALID_HANDLE	One of hResponse, hRating and hPolicy is not valid.
TWP_INVALID_PARAMETER	ppUrl is NULL.
TWP_NO_DATA	The URL rating does not violate the policy. (I.e. if TWPPolicyValidate called with hPolicy and hRating would indicate no violations).
TWP_ERROR	An internal error occurs.
TWP_NOMEM	Memory for the URL cannot be allocated.

```
TWP_RESULT TWPResponseGetUrlRatingsCount(TWPLIB_HANDLE hLib,
                                         TWPResponseHandle hResponse,
                                         unsigned int *puCount)
```

Returns the number of URL ratings currently available in the response object.

Usually, the value returned by this function will be equal to the size of the url array in the call to TWPLookupUrl.

Parameters

[in] hLib	Library handle.
[in] hResponse	Response object handle.
[out] puCount	The number of URL rating objects in the response.

Returns

TWP_INVALID_HANDLE	hResponse is not valid.
TWP_INVALID_PARAMETER	puCount is NULL.
TWP_SUCCESS	The function completed successfully.

```
TWP_RESULT TWPResponseDestroy(TWPLIB_HANDLE hLib,
                              TWPResponseHandle *phResponse)
```

Destroys and reclaims all memory from a response handle.

This function reclaims the memory used by a response handle.

Note:

This function will also discard all URL rating objects associated with this response handle.

Parameters

`[in] hLib`	Library handle.
`[in, out] phResponse`	Pointer to a response handle. When the function completes successfully, the memory pointed to by `phResponse` is set to `NULL`.

Returns

`TWP_INVALID_HANDLE`	`phResponse` is not valid.
`TWP_SUCCESS`	The handle was destroyed successfully.

Support Function

Summary

Methods	
`TWP_RESULT`	`TWPUrlRatingGetScore (TWPLIB_HANDLE hLib, TWPUrlRatingHandle hRating, int *piScore)`
	Retrieves the score assigned by security vendor to the URL.
`TWP_RESULT`	`TWPUrlRatingGetUrl (TWPLIB_HANDLE hLib, TWPUrlRatingHandle hRating, const char **ppUrl, unsigned int uiLength)`
	Retrieves the URL corresponding to the rating information represented by `hRating`.
`TWP_RESULT`	`TWPUrlRatingGetDLAUrl (TWPLIB_HANDLE hLib, TWPUrlRatingHandle hRating, const char **ppDlaUrl, unsigned int uiLength)`
	Retrieves the DLA URL.
`TWP_RESULT`	`TWPUrlRatingHasCategory (TWPLIB_HANDLE hLib, TWPUrlRatingHandle hRating, TWPCategories Category, int *piPresent)`
	Determines whether the URL rating object has the specified category.
`TWP_RESULT`	`TWPUrlRatingGetCategories (TWPLIB_HANDLE hLib, TWPUrlRatingHandle hRating, TWPCategories **ppCategory, unsigned int *puiLength)`
	Retrieves categories assigned by policy to the rated url.

Methods

```
TWP_RESULT TWPUrlRatingGetScore (TWPLIB_HANDLE hLib,
                                 TWPUrlRatingHandle hRating,
                                 int *piScore)
```

Retrieves the score assigned by security vendor to the URL.

Parameters

[in] hLib	Library handle.
[in] hRating	A URL rating handle representing the information for a URL. This handle should have been previously obtained using TWPResponseGetUrlRatingByIndex or TWPResponseGetUrlRatingByUrl.
[out] piScore	A numeric score value assigned to the URL by security vendor.

Returns

TWP_INVALID_HANDLE	hRating does not represent a valid URL rating object.
TWP_INVALID_PARAMETER	piScore is NULL.
TWP_SUCCESS	the function completes successfully and the piScore is retrieved.

```
TWP_RESULT TWPUrlRatingGetUrl (TWPLIB_HANDLE hLib,
                               TWPUrlRatingHandle hRating,
                               const char **ppUrl,
                               unsigned int uLength)
```

Retrieves the URL corresponding to the rating information represented by hRating.

Parameters

[in] hLib	Library handle.
[in] hRating	A URL rating handle representing the information for a URL. This handle should have been previously obtained using TWPResponseGetUrlRatingByIndex or TWPResponseGetUrlRatingByUrl.
[out] ppUrl	A pointer to a null terminated string representing the URL. The string is valid as long as the URL rating handle is valid.
[out] uLength	An optional pointer to the length of ppUrl.

Returns

TWP_INVALID_HANDLE	hRating does not represent a valid URL rating object.
TWP_INVALID_PARAMETER	ppUrl is NULL.
TWP_SUCCESS	the function completes successfully and the ppUrl is retrieved.

```
TWP_RESULT TWPUrlRatingGetDLAUrl (TWPLIB_HANDLE hLib,
                                  TWPUrlRatingHandle hRating,
                                  const char **ppDlaUrl,
                                  unsigned int uiLength)
```

Retrieves the DLA URL.

Parameters

[in] hLib	Library handle.
[in] hRating	A URL rating handle representing the information for a URL. This handle should have been previously obtained using TWPResponseGetUrlRatingByIndex or

	TWPResponseGetUrlRatingByUrl.
[out] ppDlaUrl	A pointer to a null terminated string representing the DLA URL. The string is valid as long as the URL rating handle is valid.
[out] uLength	An optional pointer to the length of ppDlaUrl.

Returns

TWP_INVALID_HANDLE	hRating does not represent a valid URL rating object.
TWP_INVALID_PARAMETER	ppDlaUrl is NULL.
TWP_SUCCESS	The function completes successfully and the ppDlaUrl is retrieved.

```
TWP_RESULT TWPUrlRatingGetCategory (TWPLIB_HANDLE hLib,
                                    TWPUrlRatingHandle hRating,
                                    TWPCategories Category,
                                    int *piPresent)
```

Determines whether the URL rating object has the specified category.

Parameters

[in] hLib	Library handle.
[in] hRating	A URL rating handle representing the information for a URL. This handle should have been previously obtained using TWPResponseGetUrlRatingByIndex or TWPResponseGetUrlRatingByUrl.
[in] pCategory	The category to search for in the URL rating.
[out] piPresent	Pointer to a variable that receives the category test result. A non-zero value means the category is present.

Returns

TWP_INVALID_HANDLE	hRating does not represent a valid URL rating object.
TWP_INVALID_PARAMETER	piPresent is NULL.
TWP_SUCCESS	The function completes successfully and the piPresent is set.

```
TWP_RESULT TWPUrlRatingGetCategories (TWPLIB_HANDLE hLib,
                                      TWPUrlRatingHandle hRating,
                                      TWPCategories **ppCategory,
                                      unsigned int *puiLength)
```

Retrieves categories assigned by policy to the rated url.

The function allocates memory for the array of categories. This memory must be released using TWPFnMemFree function.

Parameters

[in] hLib	Library handle.
[in] hRating	A URL rating handle representing the information for a URL. This handle should have been previously obtained using TWPResponseGetUrlRatingByIndex or TWPResponseGetUrlRatingByUrl.

`[out] ppCategory`	The pointer to a variable that will contain the address of the category array.
`[out] puiLength`	A pointer to a variable that receives the length of the category array.

Returns

`TWP_INVALID_HANDLE`	`hRating` does not represent a valid URL rating object.
`TWP_INVALID_PARAMETER`	`ppCategory or puiLength` is `NULL`.
`TWP_NOMEM`	Needed amount of memory cannot be allocated.
`TWP_SUCCESS`	The function completes successfully. If the URL rating object does not contain any categories, `puiLength` is set to `0` and `ppCategory` is set to `NULL`.

SDK Handles

Summary

Handles
`TWPConfigurationHanbdle`
Configuration object handle. A configuration object is used by other object types to alter their behaviors according to the settings passed by the application to `TWPConfigurationCreate`. An application can have multiple configuration objects. All configuration objects must be destroyed using `TWPConfigurationDestroy`.
`TWPPolicyHandle`
Policy object handle. Policy objects can be used by applications to define sets of web site categories that must be blocked. `TWPPolicyValidate` API can be used to detect that a rated URL represented by a URL rating handle violates the policy. Policy handles are created using `TWPPolicyCreate` API. Applications can have multiple policies and their number is only limited by availability of system resources. Each policy handle must be destroyed using `TWPPolicyDestroy`.
`TWPResponseHandle`
Response object handle. Response object handles are used to access information about rated URLs after the response is successfully received from the server. Response object handles can be obtained using `TWPLookupUrls`. Also, when using asynchronous requests, a response handle is passed to `TWPRequest.sendfunc` as response parameter. Response handles must be destroyed using `TWPResponseDestroy`.
`TWPUrlRatingHandle`
URL rating object handle. A URL rating object represents all information about a specific URL. An instance of URL

	rating object is created for each URL submitted for lookup in `TWPLookupUrls`. A handle to URL rating object can be obtained from a valid response handle using `TWPResponseGetUrlRatingByIndex` or `TWPResponseGetUrlRatingByUrl`. URL rating handles do not need to be destroyed explicitly. They are destroyed along with their parent response object.
`TWPLIB_HANDLE`	
	Web protection library handle. This library handle enable plug-in to support multiple instances design.

SDK Structures

Summary

TWPAPIInit	
`int`	`api_version`
	version of this structure, set to `TWP_API_VERSION`.
`TWPFnMemAlloc`	`memallocfunc`
	Pointer to malloc function.
`TWPFnMemFree`	`Memfreefunc`
	Pointer to free function.

TWPConfiguration	
`int`	`config_version`
	version of this structure, set to `TWP_CONFIG_VERSION`.
`const char *`	`client_id`
	(optional) An ID obtained from security vendor.
`const char *`	`client_key`
	(optional) A key obtained from security vendor.
`const char *`	`host`
	(optional) host name to submit requests to. Specify `NULL` to use the default host.
`Int`	`obfuscate_request`
	If not zero, the request data is obfuscated.
`TWPFnRandom`	`randomfunc`
	Pointer to a rand function that will be used to obfuscate requests (if `obfuscate_request` is zero, this field is ignored)
`int`	`secure_connection`
	Use secure connections (url of the server will start with 'https://').
`int`	`Skip_dla`
	If set to zero, DLA data is not requested from the server.

TWPRequest	
int	request_version
	version of this structure, set to TWP_REQUEST_VERSION.
TWPFnRequestSetUrl	seturlfunc
	Pointer to a function that allows to set request URL.
TWPFnRequestSetMethod	setmethodfunc
	Pointer to a function that allows to set the method of the request.
TWPFnRequestSend	sendfunc
	Pointer to a function that submits the request to the server.
TWPFnRequestReceive	receivefunc
	Pointer to a function that receives the data from the server. If it is NULL, the request is assumed to be asynchronous.

SDK Types

Summary

TWPAPIInit	
int	TWP_RESULT
	The return type of all Web Protection Services APIs.
void *	(*TWPFnMemAlloc)(TWPMallocSizeT Size)
	Memory allocation routine type.
void	(*TWPFnMemFree)(void *pAddress)
	Memory deallocation routine type.
long	(*TWPFnRandom)(void)
	Randomization routine type.
TWP_RESULT	(*TWPFnRequestSetUrl)(TWPRequest *pRequest, const char *pUrl, unsigned int uLength)
	Request interface: 'set request url' routine type.
TWP_RESULT	(*TWPFnRequestSetMethod)(TWPRequest *pRequest, TWPSubmitMethod Method)
	Request interface: 'set request method' routine type.
TWP_RESULT	(*TWPFnRequestSend)(TWPRequest *pRequest, TWPResponseHandle hResponse, const void *pData, unsigned int uLength)
	Request interface: 'send request' routine type.
TWP_RESULT	(*TWPFnRequestReceive)(TWPRequest *pRequest, void *pBuffer, unsigned int uBufferLength, unsigned int *puLength)
	Request interface: 'receive' routine type.

```
void*(* TWPFnMemAlloc)(TWPMallocSizeT Size)
```

Memory allocation routine type.

Parameters

[in] Size	The amount of memory to allocate in bytes. TWPMallocSizeT is defined as size_t on most platforms.

Returns

void *	Pointer to a newly allocated memory or NULL in cases of failure.

```
void(*TWPFnMemFree)(void *pAddress)
```

Memory deallocation routine type.

Parameters

[in] pAddress	Pointer to a memory region being freed.

Returns

None.

```
long (*TWPFnRandom)(void)
```

Randomization routine type.

Parameters

None.

Returns

a random value.

```
TWP_RESULT (*TWPFnRequestSetUrl)(TWPRequest *pRequest,
                                 const char *pUrl,
                                 unsigned int uLength)
```

Request interface: 'receive' routine type.

Parameters

[in] pRequest	Pointer to an instance of TWPRequest that was passed to TWPLookupUrls call.
[in] pUrl	A URL of the new request.
[in] uLength	The length of the URL.

Returns

TWP_SUCCESS	if request URL was set successfully.
Other	Predefined error codes or a custom return code. This value will be returned from TWPLookupUrls to the application.

```
TWP_RESULT (*TWPFnRequestSetMethod)(TWPRequest *pRequest,
                                    TWPSubmitMethod Method)
```

Request interface: 'set request method' routine type.

Parameters

[in] pRequest	Pointer to an instance of TWPRequest that was passed to TWPLookupUrls call.

[in] Method	Currently, only WP_POST is passed.

Returns

TWP_SUCCESS	if request URL was set successfully.
Other	Predefined error codes or a custom return code. This value will be returned from TWPLookupUrls to the application.

```
TWP_RESULT (*TWPFnRequestSend)(TWPRequest *pRequest,
                               TWPResponseHandle hResponse,
                               const void *pData,
                               unsigned int uLength)
```

Request interface: 'send request' routine type.

Parameters

[in] pRequest	Pointer to an instance of TWPRequest that was passed to TWPLookupUrls call.
[in] hResponse	For synchronous requests is NULL. For asynchronous requests, an uninitialized response object that has to be passed to TWPResponseWrite.
[in] pData	request body to send to the server.
[in] uLength	The length of the data.

Returns

TWP_SUCCESS	if request URL was set successfully.
Other	Predefined error codes or a custom return code. This value will be returned from TWPLookupUrls to the application.

```
TWP_RESULT (*TWPFnRequestReceive)(TWPRequest *pRequest,
                                  void *pBuffer,
                                  unsigned int uBufferLength,
                                  unsigned int *puLength)
```

Request interface: 'receive' routine type.

Parameters

[in] pRequest	Pointer to an instance of TWPRequest that was passed to TWPLookupUrls call.
[in] pBuffer	a buffer where the response is to be stored.
[in] uBufferLength	the length of the buffer in bytes.
[in] puLength	number of bytes actually stored in the buffer..

Returns

TWP_SUCCESS	if request URL was set successfully.
Other	Predefined error codes or a custom return code. This value will be returned from TWPLookupUrls to the application.

SDK Enums

Summary

Enums
`TWPCategories`
Defines categories assigned by security vendor to websites. Applications can use these categories in defining policies that they want to enforce (such as, "block porn, nudity & gambling sites").
`TWPResultCodes`
This enumeration is used to declare numeric result codes for SDK functions. The type itself is not used in any other context in this SDK.
`TWPRiskLevel`
Defines the risk levels that users visiting the URL are exposed to. Each risk level corresponds to a specific range of rating scores returned by TWPUrlRatingGetScore.
`TWPScoreRange`
Defines rating score thresholds for determining a TWPRiskLevel of a rated URL. The score ranges defined below are recommended by security vendor, but application developers can use custom ranges for determining risk levels.
`TWPSubmitMethod`
Defines possible values for the `method` parameter in TWPFnRequestSetMethod. Currently only "POST" method for sending HTTP/S requests is supported.

Submit method

`WP_POST`	use "POST" method for sending HTTP/S requests to the server.

Score range

`MinimalLow`	Lowest score corresponding to the Minimal level.
`MinimalHigh`	Highest score corresponding to the Minimal level.
`UnverifiedLow`	Lowest score corresponding to the Unverified level.

UnverifiedHigh	Highest score corresponding to the Unverified level.
MediumLow	Lowest score corresponding to the Medium level.
MediumHigh	Highest score corresponding to the Medium level.
HighLow	Lowest score corresponding to the High level.
HighHigh	Highest score corresponding to the High level.

Risk level

Minimal	The webpage most likely belongs to a legitimate web server, but all content needs further inspection.
Unverified	The webpage may belong to a legitimate web server, but displays a few properties suggesting that further inspection is needed.
Medium	The webpage shows many characteristics associated with a malicious web server and any content from it needs special scrutiny.
High	The web page contains malicious content or is not from a legitimate host server.

Result codes

TWP_SUCCESS	Function completed successfully.
TWP_ERROR	Function completed with error.
TWP_NOMEM	The required amount of memory cannot be allocated. This usually happens when WP_FnMemAlloc function returns NULL.
TWP_INVALID_HANDLE	An invalid handle was provided as an input parameter to a function.
TWP_INVALID_PARAMETER	An invalid parameter was provided to a function.
TWP_INVALID_VERSION	The structure version is incorrect. Developers should use corresponding structure versions defined in wp_api.h.
TWP_INVALID_RESPONSE	The server response is malformed.
TWP_NO_DATA	Requested data is not available. This error code is usually returned when a non-existent URL rating object is being accessed.

Categories definitions

Category Name	Description
Artcultureheritage	Web pages that contain virtual art galleries, artist sites (including sculpture and photography), museums, ethnic customs, and country customs. This category does not include online photograph albums. See the Media Sharing category.
Alcohol	Web pages that mainly sell, promote, or advocate the use of alcohol, such as beer, wine, and liquor. This category also includes cocktail recipes and home-brewing instructions

Anonymizers	Web pages that purposefully allow users to browse the web by hiding their IP address, or other personal identification information, in order to bypass local filtering policies and access any web page. Anonymizer web pages also block any tracking technologies, such as cookies or browser history. Some methods also prevent OS version and web page history from being forwarded to the web page.
Anonymizingutilities	Web pages that result in anonymous web browsing without the explicit intent to provide such a service.
Business	Web pages that provide business-related information, such as corporate overviews or business planning and strategies.
Chat	Web pages that provide web-based, real-time social messaging in public and private chat rooms.
Publicinformation	Web pages that provide general reference information such as public service providers, regional information, transportation schedules, maps, or weather reports.
Potentialcriminalactivities	Web pages, which security vendor believes provide instructions to commit illegal or criminal activities. Instructions include committing murder or suicide, sabotage, bomb-making, lock-picking, service theft, evading law enforcement, or spoofing drug tests. This category might also include information on how to distribute illegal content, perpetrate fraud, or consumer scams. This category does not include computer-related fraud. See the Potential Hacking/Computer Crime category.
Drugs	Websites that provide information on the purchase, manufacture, and use of illegal or recreational drugs. This category includes displaying, selling, or describing the use of drug paraphernalia, and tips on legal highs, such as glue sniffing, the misuse of prescription drugs, or the abuse of other legal substances. This category does not include sites with exclusive health or political themes.
Educationreference	Web pages devoted to academic-related content such as academic subjects (mathematics, history), school or university web pages, and education administration pages (school boards, teacher curriculum).
Entertainment	Web pages that provide information about cinema, theater, music, television, infotainment, entertainment industry gossip-news, and sites about celebrities such as actors and musicians. This category also includes sites where the content is devoted to providing entertainment on the web, such as horoscopes or fan clubs.
Extreme	Web pages that provide content considered gory, perverse, or horrific. This category is used with the categories of Gruesome Content, Discrimination, Pornography, Violence, or Game/Cartoon Violence to identify URLs that are at the extremes of these categories. An example is illegal pornography, which is in the categories of Pornography, Potential Criminal Activities, and Extreme.
Financebanking	Web pages that provide financial information or access to online financial accounts. This category includes web pages that provide financial information include news or services that deal with the management of money, but do not provide access to sensitive financial account information, such as financial statistics, or consultation in areas such as taxes, mortgage, insurance, accreditation, or investment. Access to online financial accounts includes personal or business related banking, money management, tax consulting, mutual funds, credit cards, credit unions, insurance, other financial publications and services. This category includes stock information (but not stock trading), home finance, and government-related financial information.
Gambling	Web pages that allow users to wager or place bets online, or provide gambling software that allows online betting, such as casino games, betting pools, sports betting, and lotteries. This category includes lotteries if you can buy the tickets or play online. This category does not include web pages related to gambling that do not allow betting online. See the Gambling Related category. Financial spread betting is in the Stock Trading category.
Games	Web pages that offer online games and related information such as cheats, codes, demos, emulators, online contests or role-playing games, gaming clans, game manufacturer sites, fantasy or virtual sports leagues, and other gaming sites without chances of profit. This category includes gaming consoles, such as Microsoft Xbox(R) and Sony PlayStation(R).
Governmentmilitary	Web pages that contain content maintained by governmental or military organizations, such as government branches or agencies, police departments, fire departments, civil defense, counterterrorism organizations, or supranational organizations, such as the United Nations or the European Union. This category includes military and veterans' medical facilities.
Potentialhackingcomputercrime	Web pages, which security vendor believes provide instructions, or otherwise enable, fraud, crime, or malicious activity that is computer-oriented. This category includes web pages related to computer crime include malicious hacking information or tools that help individuals gain unauthorized access to computers and networks (root kits, kiddy scripts). This category also includes other areas of electronic fraud such as dialer scams and illegal manipulation of electronic devices. This category does not include illegal software. See Potential Illegal Software.
Health	Web pages that cover all health-related information and health care services. Health information includes topics to improve an individual's well-being, whether for physical or mental health (diet, nutrition, fitness, or parenting). Health care services include

	providers such as health insurance, hospitals, clinics, and independent physicians. This category does not include cosmetic surgery, marketing/selling pharmaceuticals, or animal-related medical services.
Humorcomics	Web pages that provide comical or funny content. This category includes sites with jokes, sketches, comics, and satire pages. This category might also include graphic novel content, which is often associated with comics.
Discrimination	Web pages, which security vendor believes provide information that explicitly encourages the oppression or discrimination of a specific group of individuals. This category includes promoting an agenda against groups based on race, religion, nationality, gender, age, disability, or sexual orientation. This category might include political parties and organizations with a specific hate-based agenda, and cyber-bullying where it is extreme. This category does not include jokes and humor, unless the focus of the entire site is considered discriminatory.
Instantmessaging	Web pages that provide software for real-time communication over a network exclusively for users who joined a member's contact list or an instant-messaging session. Most instant-messaging software includes features such as file transfer, PC-to-PC phone calls, and can track when other people log on and off.
Stocktrading	Web pages that offer purchasing, selling, or trading of shares online. This category also includes ticker-tape information that enables viewing of real-time stock prices and financial spread betting in the stock market. Other betting is in the Gambling category. This category does not include sites that offer information about stocks, but do not offer purchasing, selling, or trading of shares. See the Finance/Banking category.
Internetradiotv	Web pages that provide software or access to continuous audio or video broadcasting, such as Internet radio, TV programming, or podcasting. Quick downloads and shorter streams that consume less bandwidth are in the Streaming Media or Media Downloads categories.
Jobsearch	Web pages related to a job search including sites concerned with resume writing, interviewing, changing careers, classified advertising, and large job databases. This category also includes corporate web pages that list job openings, salary comparison sites, temporary employment, and company job-posting sites. This category does not include make-money-at-home sites. See in the Business or Online Shopping categories.
Informationsecurity	Web pages that legitimately provide information about data protection. This category includes detailed information for safeguarding business or personal data, intellectual property, privacy, and infrastructure on the Internet, private networks, or in other bandwidth services such as telecommunications. This category does not include: o Legitimate information security companies and security software providers, such as virus protection companies. o Sites that intend to exploit security or teach how to bypass security. See the Potential Hacking/Computer Crime category.
E_RESERVED_1	Not used.
Mobilephone	Web pages that sell media, software, or utilities for mobile phones that can be downloaded and delivered to mobile phones. Examples include ringtones, logos/skins, games, screen-savers, text-based tunes, and software for SMS, MMS, WAP, and other mobile phone protocols.
Mediadownloads	Web pages that provide audio or video files for download such as MP3, WAV, AVI, and MPEG formats. The files are saved to, and played from, the user's computer. This category does not include audio or video files that are played directly through a browser window.
Malicioussites	Web pages that deploy code designed specifically to hijack your computer's settings or activity. This category includes self-installing applications (called "drive-by" executable file downloads), Trojan horses, and viruses that exploit security vulnerabilities in browsers or other applications. A web page can later lose its malicious status and be moved to another category. This category does not include spyware/adware because of its intent. Spyware/adware covertly gathers and sends information to another party. Malicious sites intend to take action on another computer to cause damage. This category does not include browser exploits and malicious downloads, which are in separate categories.
E_RESERVED_2	Not used.
Nudity	Web pages that have non-pornographic images of the bare human body. This category includes classic sculpture and paintings, artistic nude photographs, some naturism pictures, and detailed medical illustrations. This category does not include high-profile sites where nudity is not a concern for visitors.
Nonprofitadvocacyngo	Web pages from charitable or educational groups that fulfill a stated mission, benefiting the larger community, such as clubs, lobbies, communities, non-profit organizations, labor unions, and advocacy groups. Examples are Masons, Elks, Boy and Girl Scouts, or Big Brothers. This category includes organizations with governmental endorsements, but not government-run organizations.
Generalnews	Web pages that provide online news media, such as international or regional news broadcasting and publication. This category includes portal sites that provide news content.
Onlineshopping	Web pages that sell products or services online. Web pages selling a broad range of products might pose a risk to users by offering access to items that are normally in

	other categories such as Pornography, Weapons, Nudity, or Violence. Web pages selling such content exclusively are in their respective categories.
Provocativeattire	Web pages with pictures that include alluring or revealing attire, lingerie and swimsuits, or supermodel or celebrity photograph collections, but do not involve nudity. This category does not include sites with swimwear or similar attire that is not intended to be provocative. For example, Olympic swimming sites are not in this category.
P2pfilesharing	Web pages that allow the exchange of files between computers and users for business or personal use, such as downloadable music. P2P clients allow users to search for and exchange files from a peer-user network. They often include spyware or real-time chat capabilities. This category includes BitTorrent web pages.
Politicsopinion	Web pages covering political parties, individuals in political life, and opinion on various topics. This category might also cover laws and political opinion about drugs. This category includes URLs for political parties, political campaigning, and opinions on various topics, including political debates.
Personalpages	Personal home pages that share a common domain such as those hosted by ISPs, university/education servers, or free web page hosts. This category also includes unique domains that contain personal information, such as a personal home page. This category does not include home pages of public figures. Those web pages are in categories such as Entertainment or Sports. However, personal pages do not have high traffic, making them difficult to categorize.
Portalsites	Web pages that serve as major gateways or directories to content on the web. Many portal sites also provide a variety of internal site features or services such as search engines, email, news, and entertainment. Mailing list sites with a variety of content are in this category. This category does not include sites with topic-specific content. Those sites are in more specific categories, such as Entertainment.
Remoteaccess	Web pages that provide remote access to a program, online service, or an entire computer system. Although remote access is often used legitimately to run a computer from a remote location, it creates a security risk, such as backdoor access. Backdoor access, written by the original programmer, allows the system to be controlled by another party without the user's knowledge.
Religionideology	Web pages with content related to religious topics and beliefs in human spirituality that are not within the major religions. This category includes religious discussion, beliefs, articles, and information for local congregations or groups such as a church homepage, unless the site is already in the Majorglobalreligions category. This category also includes comparative religion, or sites that include religions and ideologies. This category does not include astrology and horoscope sites. Traditional, popular, global religions are in the Majorglobalreligions category.
Resourcesharing	Web pages that harness idle or unused computer resources to focus on a common task. The task can be on a company or an international basis. Well known examples are the SETI program and the Human Genome Project, which use the idle time of thousands of volunteered computers to analyze data.
Searchengines	Web pages that provide search results that enable users to find information on the Internet based on key words. This category does not include site-specific search engines.
Sports	Web pages related to professional or organized recreational sports. This category includes sporting news, events, and information such as playing tips, strategies, game scores, or player trades. This category does not include fantasy leagues, sports centers, athletic clubs, fitness or martial arts clubs, and non-league billiards, darts, or other such activities.
Streamingmedia	Web pages that provide streaming media, or contain software plug-ins for displaying audio and visual data before the entire file has been transmitted. This category does not include audio or video files that are downloaded to a user's computer before being played.
Sharewarefreeware	Web pages that are repositories of downloadable copies of shareware and freeware. Shareware is distributed on an honor system by which the shareware is delivered free of charge, but the author requests a small fee if the program is suitable and is regularly used. Freeware is software that is available without any cost. This category does not include subscription-based software.
Pornography	Web pages, which security vendor believes contain materials intended to be sexually arousing or erotic. This category includes fetish pages, animation, cartoons, stories, and illegal pornography.
Spywareadwarekeyloggers	Web pages that violate personal or corporate privacy and security. This category includes:
	- URLs that download software or execute programs that gather user information and send it to a third party without the user's explicit knowledge or consent.
	- Software that monitors or tracks activity on others' computer by recording the key strokes (Keyloggers).
	- Sites that distribute software known to contain spyware or adware.
	This category is different from the Malicious Sites category because of its intent. Spyware and keyloggers covertly gather and send information to another party. A

	malicious site intends to install software or take an action without user approval or knowledge. This category does not include web pages that download Potentially Unwanted Programs (PUPs).
Tobacco	Web pages that sell, promote, or advocate the use of tobacco products, tobacco paraphernalia, including cigarettes, cigars, pipes, snuff and chewing tobacco.
Travel	Web pages that promote personal or business travel, such as hotels, resorts, airlines, ground transportation, car rentals, travel agencies, and general tourist and travel information. This category also includes sites for buying tickets or accommodation. This category does not include personal vacation photographs. Although tourism information might be on governmental sites, such web pages are not in this category.
Violence	Web pages that contain real or lifelike images or text that portray, describe, or advocate physical assaults against people, animals, or institutions, such as depictions of war, suicide, mutilation, or dismemberment.
Webads	Web pages that provide advertisement-hosting or programs that create advertisements. Examples include links, source code or applets for banners, popups, and other kinds of static or dynamically generated advertisements that appear on web pages. This category is intended to block advertisements on web pages, not the companies that provide the advertisements or advertising services. This category does not include aggressive advertising adware.
Weapons	Web pages that provide information about buying, making, modifying, or using weapons, such as guns, knives, swords, paintball guns, and ammunition, explosives, and weapon accessories. This category also includes sites that contain content for:
	- Weapons for personal or military use.
	- Homemade weapons.
	- Non-lethal weapons, such as mace, pepper spray, or Taser guns.
	- Weapons facilities, such as shooting ranges.
	- Government or military oriented weapons.
	This category does not include political action groups, such as the NRA.
Webmail	Web pages that enable users to send or receive email through the Internet.
Webphone	Web pages that enable users to make telephone calls via the Internet or obtain information or software for this purpose. Web Phone service is also called Internet Telephony, or VoIP. Web phone service includes PC-to-PC, PC-to-phone, and phone-to-phone services connecting via TCP/IP networks.
Auctionsclassifieds	Web pages that provide online bidding and selling of items or services. This category includes web pages that focus on bidding and sales. This category does not include classified advertisements such as real estate postings, personal ads, or companies marketing their auctions.
Forumbulletinboards	Web pages that provide access (http://) to Usenet newsgroups or hold discussions and post user-generated content, such as real-time message posting for an interest group. This category also includes archives of files uploaded to newsgroups. This category does not include message forums with a business or technical support focus.
Profanity	Web pages that contain crude, vulgar, or obscene language or gestures.
Schoolcheatinginformation	Web pages that promote plagiarism or cheating by providing free or fee-based term papers, written essays, or exam answers. This category does not include sites that offer student help, discuss literature, films, or books, or other content that is often the subject of research papers.
Sexualmaterials	Web pages that describe or depict sexual acts, but are not intended to be arousing or erotic. Examples of sexual materials include sex education, sexual innuendo, humor, or sex related merchandise. This category does not include web pages with content intended to arouse.
Gruesomecontent	Web pages with content that can be considered tasteless, gross, shocking, or gruesome. This category includes tasteless humor, bodily excretory functions, graphic medical or accident scene photographs (containing blood or wounds), extreme forms of body modification (cutting, branding, or genital piercing), or shocking depictions of inhumane animal treatment. This category does not include web pages with content pertaining to physical assault.
Visualsearchengine	Web pages that provide image-specific search results such as thumbnail pictures. This category does not include sites that offer site-specific visual search engines.
Technicalbusinessforums	Web pages with a technical or business focus that provide online message posting or real-time chatting, such as technical support or interactive business communication. Although users can post any type of content, these forums tend to present less risk of containing offensive content. Sites that offer a variety of forums with themes, including technical and business content, are only in the categories of Forum/Bulletin Boards or Chat. Forums with a primary technical or business focus only are in the Technical/Business Forums category.
Gamblingrelated	Web pages that offer information about gambling, without providing the means to gamble. This category includes casino-related web pages that do not offer online gambling, gambling links, tips, sports picks, lottery results, and horse, car, or boat racing.
Messaging	Web pages that provide text messaging. Examples include text messaging to mobile phones, PDAs, fax machines, and internal website user-to-user messaging or

	site-to-site messaging. This category does not include real-time chat or instant messaging, or message posts that can be viewed by anyone but the intended recipient. More complex messaging features that allow file attachments, include larger storage and bandwidth use, or can receive external site spam are in the Web Mail category.
Gamecartoonviolence	Web pages that provide fantasy or fictitious representations of violence within the context of games, comics, cartoons, or graphic novels. This category includes images and textual descriptions of physical assaults or hand-to-hand combat, and grave injury and destruction caused by weapons or explosives.
Phishing	Links to web pages that typically arrive in hoax email, established to steal user account information. These sites falsely represent themselves and appear as legitimate company web pages in order to deceive and obtain user account information for perpetrating fraud or theft.
Personalnetworkstorage	Web pages that allow users to upload folders and files to an online network server in order to backup, share, edit, or retrieve files or folders from any web browser.
Spamurls	Links to web pages that arrive in unsolicited spam email. Spam URL content ranges from product marketing to potentially offensive or fraudulent sites. This category includes sites built only for spamming purposes such as spam blogs or comment spam. This category does not include links to web pages often seen in hoax email, where the web pages are designed to steal user account information.
Interactivewebapplications	Web pages that provide access to live or interactive web applications, such as browser-based office suites and groupware. This category includes sites with business, academic, or individual focus. This category does not include sites providing access to interactive web applications that do not take critical user data or offer security risks, such as Google Maps. Sites that focus primarily on web meetings for business use are in the Web Meetings category.
Fashionbeauty	Web pages that market clothing, cosmetics, jewelry, and other fashion-oriented products, accessories, or services. This category also includes product reviews, comparisons, and general consumer information, and services such as hair salons, tanning salons, tattoo studios, and body-piercing studios. This category does not include fashion-related content such as modeling or celebrity fashion unless the site focuses on marketing the product line.
Softwarehardware	Web pages related to computing software and hardware, including vendors, product marketing and reviews, deployment and maintenance of software and hardware, and software updates and add-ons such as scripts, plug-ins, or drivers. Hardware includes computer parts, accessories, and electronic equipment used with computers and networks. This category includes the marketing of software and hardware, and magazines focused on software or hardware product reviews or industry trends.
Potentialillegalsoftware	Web pages, which security vendor believes offer information to potentially 'pirated' or illegally distribute software or electronic media, such as copyrighted music or film, distribution of illegal license key generators, software cracks, and serial numbers. This category does not include peer-to-peer web pages.
Contentserver	URLs for servers that host images, media files, or JavaScript for one or more sites and are intended to speed up content retrieval for existing web servers, such as Apache. Content servers generally do not have content posted or through-site navigation for web surfers. Content servers hosting images do not allow users to browse the photographs. This category includes domain-level and sub-domain-level URLs that function as content servers.
Internetservices	Web pages that provide services for publication and maintenance of Internet sites such as web design, domain registration, Internet Service Providers, and broadband and telecommunications companies that provide web services. This category includes web utilities such as statistics and access logs, and web graphics like clip art.
Mediasharing	Web pages that allow users to upload, search for, and share media files and photographs, such as online photograph albums.
Incidentalnudity	Web pages that contain non-pornographic images of the bare human body like those in classic sculpture and paintings, or medical images. This category enables you to allow or block sites in order to address cultural or geographic differences in opinion about nudity. For example, you can use this category to block access to nudity, but allow access when nudity is not the primary focus of a site, such as news sites or major portals.
Marketingmerchandising	Web pages that promote individual or business products or services on the web, but do not sell their products or services online. This category includes websites that are generally a company overview, describing services or products that cannot be purchased directly from these sites. Examples include automobile manufacturer sites, wedding photography services, or graphic design services. This category does not include:
	- Other categories that imply marketing such as Alcohol, Auctions/Classifieds, Drugs, Finance/Banking, Mobile Phone, Online Shopping, Real Estate, School Cheating Information, Software/Hardware, Stock Trading, Tobacco, Travel, and Weapons.
	- Sites that market their services only to other businesses.
	- Sites that rob or cheat consumers.
Parkeddomain	Web pages that once served content, but their domains have been sold or abandoned

	and are no longer registered. Parked domains do not host their own content, but usually redirect users to a generic page that states the domain name is for sale, or redirect users to a generic search engine and portal page, some of which provide valid search engine results.
Pharmacy	Web pages that provide reviews, descriptions, and market or sell prescription-based drugs, over-the-counter drugs, birth control, or dietary supplements.
Restaurants	Web pages that provide information about restaurants, bars, catering, take-out and delivery, including online ordering. This category includes sites that provide information about location, hours, prices, menus and related dietary information. This category also includes restaurant guides and reviews, and cafes and coffee shops. This category does not include groceries, wholesale food, non-profit and charitable food organizations, or bars that do not focus on serving food.
Realestate	Web pages that provide commercial or residential real estate services and information. Service and information includes sales and rental of living space or retail space and guides for apartments, housing, and property, and information on appraisal and brokerage. This category includes sites that allow you to browse model homes. This category does not include content related to personal finance, such as credit applications.
Recreationhobbies	Web pages for recreational organizations and facilities that include content devoted to recreational activities and hobbies. This category includes information about public swimming pools, zoos, fairs, festivals, amusement parks, recreation guides, hiking, fishing, bird watching, or stamp collecting. This category does not include activities that need no active participation, such as watching a movie or reading celebrity gossip.
Blogswiki	Web pages containing dynamic content, which often changes because users can post or edit content at any time. This category covers the risks with dynamic content that might range from harmless to offensive.
Digitalpostcards	Web pages that allow people to send and receive digital postcards and greeting cards via the Internet. Digital postcards give messages of greeting or sentiment, usually for special occasions such as birthdays and holidays.
Historicalrevisionism	Web pages that denounce, or offer different interpretations of, significant historical facts, such as holocaust denial. This category includes sites that might have a legitimate academic value, however many of these sites include illegitimate manipulation of history for political purposes. This category does not include all re-examination of historical facts, only historical events that are highly sensitive.
Technicalinformation	Web pages that provide computing information with an educational focus in areas such as Information Technology, computer programming, and certification. Examples include Linux user groups, UNIX commands, software tutorials, or dictionaries of technical terms. Most sites in this category might be subdirectories of larger domains. For example, a software site with a tutorial page is in this category only at the tutorial page URL. This category does not include content about information security.
Datingpersonals	Web pages that provide networking for online dating, matchmaking, escort services, or introductions to potential spouses. This category includes sites that provide personal or group profiles, and allow their members to interact through real-time communication, message posting, public bulletins, and media sharing. This category does not include sites that provide social networking that might include dating, but are not specific to dating.
Motorvehicles	Websites for manufacturers and dealerships of consumer transportation vehicles, such as cars, vans, trucks, SUVs, motorcycles, and scooters. This category also includes sites that provide product marketing, reviews, comparisons, pricing information, auto fairs, auto expos, and general consumer information about motor vehicles. This category does not include automotive accessories, mechanics, auto-body shops, and recreational hobby pages. This category does not include sites that provide business-to-business-only content regarding motor vehicles.
Professionalnetworking	Web pages that provide social networking exclusively for professional or business purposes. This category includes sites that provide personal or group profiles, and enable their members to interact through real-time communication, message posting, public bulletins, and media sharing. This category also contains alumni sites that have a networking function. This category does not include social networking sites where the focus might vary, but include friendship, dating, or professional focuses.
Socialnetworking	Web pages that enable social networking for a variety of purposes, such as friendship, dating, professional, or topics of interest. These sites provide personal or group profiles and enable interaction among their members through real-time communication, message posting, public bulletins, and media sharing. This category does not include sites that are exclusive to dating, matchmaking, or a specific professional networking focus.
Texttranslators	Web pages that allow users to type phrases or a block of text to translate it from one language into another. This category also includes language identifier web pages. This category does not include URL translation, which is a different activity. URL translation translates the content of a web page, not a word, phrase, or block of text typed into an input field. URL translation is in the Anonymizing Utilities category.
Webmeetings	Web pages that host live meetings, video conferences, and interactive presentations

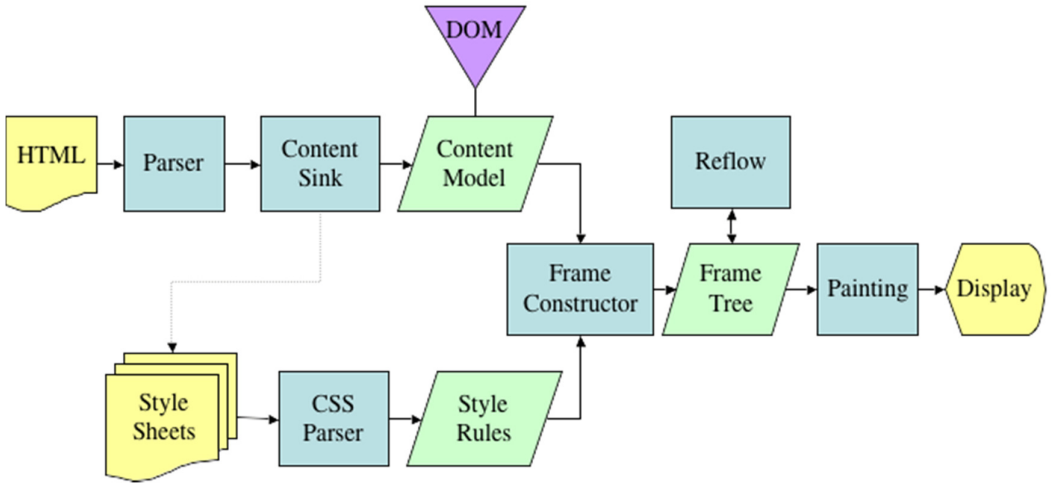
	mainly for businesses. Web meetings generally include streaming audio and video, and allow data transfer or office-oriented application sharing, such as online presentations. For web-meeting sites that offer features not used in the web meeting, such as remote access or browser-based office suites, the individual URLs are in various categories.
Forkids	Web pages that are family-safe, specifically for children of approximate ages ten and under.
E_RESERVED_3	Not used.
Moderated	Bulletin boards, chat rooms, search engines, or web mail sites that are monitored by an individual or group who has the authority to block messages or content considered inappropriate. This category does not include sites with posted rules against offensive content.
Textspokenonly	Content that is text or audio only, and does not contain pictures. This category can be used as an exception to allow explicit text and recorded material to be accessed when you want pictures blocked using the Pornography, Violence, or Sexual Materials categories. Libraries or universities can use this category to prevent the display of offensive graphics in their public facilities.
Controversialopinions	Web pages that contain opinions that are likely to offend political or social sensibilities and incite controversy. Much of this content is at the extremes of public opinion. Examples include suicide pacts, pro-anorexia, xenophobic, ethnocentric, fundamentalist viewpoints, disinformation, or critical examination of one group of people.
Residentialipaddresses	IP addresses (and any domains associated with them) that access the Internet by DSL modems or cable modems. Because this content is not generally intended for Internet access via HTTP, access to the Internet through these IP addresses can indicate suspicious behavior. This behavior might be related to malware located on the home computer or homegrown gateways set up to allow anonymous Internet access.
Browserexploits	Web pages containing browser exploits. A browser exploit is a piece of code that exploits a software bug in a web browser so that the code makes the browser do something unexpected, including stop running, read or write local files, propagate a virus or install spyware. Malicious code may exploit HTML, JavaScript, Images, ActiveX, Java and other web technologies. Exploits, sometimes called drive-by-downloads, can install themselves on unprotected computers often without a consumer's knowledge, where the code can install keystroke loggers (to steal passwords) and Trojan programs (to turn a computer into a "bot" or "slave machine"). This category includes URLs that distribute or execute any browser exploit.
Consumerprotection	Websites that try to rob or cheat consumers. Some examples of their activities include selling counterfeit products, selling products that were originally provided for free, or improperly using the brand of another company. This category includes websites that exploit kindness or ignorance such as fraudulently collecting money for a charity, or running fraudulent investment schemes. This category also includes sites where many consumers reported being cheated or not receiving services. This category does not include phishing, which tries to perpetrate fraud or theft by stealing account information.
Illegaluk	Web pages that contain child sexual abuse content hosted anywhere in the world, and criminally obscene and incitement to racial hatred content hosted in the UK. This category contains URLs provided by the Internet Watch Foundation (IWF) to block access to sites including those that are detrimental to child welfare. The IWF website is at http://www.iwf.org.uk.
Majorglobalreligions	Web pages with content about religious topics and information related to major religions. This category includes sites that cover religious content such as discussion, beliefs, non-controversial commentary, articles, and information for local congregations such as a church or synagogue homepage. The religions in this category are Baha'I, Buddhism, Chinese Traditional, Christianity, Hinduism, Islam, Jainism, Judaism, Shinto, Sikhism, Tenrikyo, and Zoroastrianism. This category does not include sites with content about other religions and ideologies, such as comparative religion.
Maliciousdownloads	Web pages that allow a user to inadvertently download code that is harmful or annoying. This category includes downloadable files like screensavers, toolbars and file-sharing programs that contain adware, spyware, viruses and other malicious computer code. Sometimes, the malware is added without the user's knowledge, as when they click "Yes" or "I agree" without reading the full terms and conditions. The effects can include slower computer speeds, theft of passwords, and the loss or damage of valuable personal files.
Potentiallyunwantedprograms	Web pages that contain Potentially Unwanted Programs (PUPs). PUPs are often made for a beneficial purpose but they alter the security of a computer or the computer user's privacy. Computer users who are concerned about security or privacy might want to be informed about this software, and in some cases, they might want to remove this software from their computers.
LastCategoryPlaceholder	This value is for internal use only!
OverallPhishing	The rating score for this site is negative (meaning 'phishing'). This value can be used when only a score is returned by the server without categories.

OverallRiskHigh	The rating score corresponds to a 'Red' site.
OverallRiskMedium	The rating score corresponds to a 'Yellow' site.
OverallRiskMinimal	The rating score corresponds to a 'Green' site.
OverallRiskUnverified	The rating score corresponds to an 'Unverified' site.
LastAttributePlaceholder	This value is for internal use only!

Implementation Guide

Security Browser

This is a typical browser rendering process. Web protection API is to enable caller to determine if the URL is good for rendering according to caller's settings (categories).



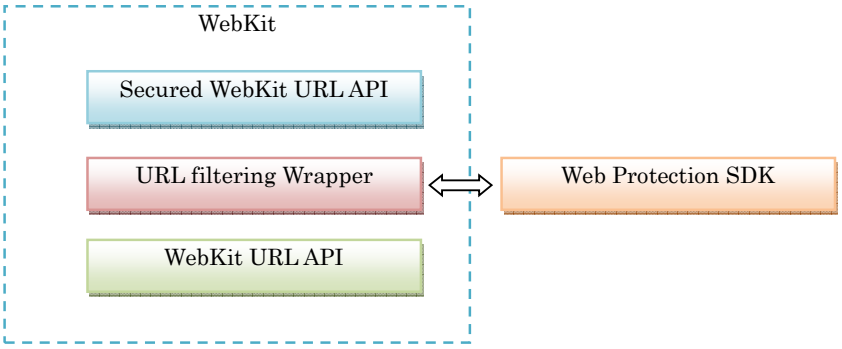
Here is an example of security enhanced WebView function call:

```
void webkit_web_view_load_html_string(WebKitWebView *web_view,
                                      const gchar *content,
                                      const gchar *base_uri) {
    const char *Urls[1];
    TWPResponseHandle hResponse;
    TWPRating hRating;
    int violated = 0;

    Urls[0] = asprintf(&Urls[0], "%s/%s", base_url, content);
    TWPLookupUrls(mTwpConfig, mRequest, 0, Urls, 1, &hResponse);
    TWPResponseGetUrlRatingByIndex(hResponse, 0, &hRating);
    TWPPolicyValidate(mPolicy, hRating, &violated);
    if (violated) {
       show_block_page();
    } else {
       Original_webkit_web_view_load_html_string(web_view,
                                                 content,
                                                 base_uri);
    }
```

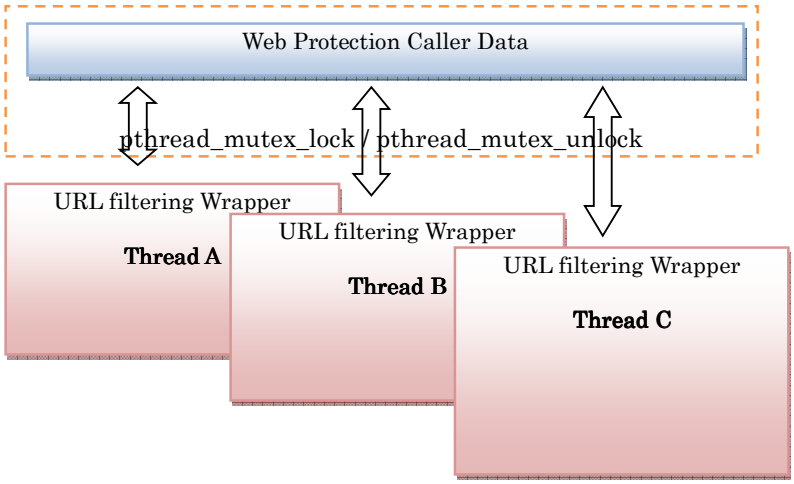
Reference design

The design for security browser should be hook oriented. The browser engine should be able to block the web access at the time that the URL conflict with the security policy. Basically implementer need to come up with the API list which need to be hooked and replace them with a list of security enhanced function call. For example, one of the WebKit API named webkit_web_view_load_html_string() and we decide to hook this call, we can rename it to: original_webkit_web_view_load_html_string(), and change the implementation of the API to be wrapped with web protection URL filtering check.



Thread safe consideration

Since web protection SDK requires caller to create configurations, policies as well as library handles to drive the URL filtering functions. Usually, browser needs to handle those requests in different thread but sharing the configurations, policies and handles. Then we have to take thread safe into our design consideration. As Linux program, pthread library provide us a set of API with condition and mutex which can perfectly give us solution for multi-tasking data sharing. We need to protect the web protection SDK caller data (configurations, policies and library handles) with mutex and condition variables.



URL filtering implement steps

1. Initialize web protection library handle.
2. Create web protection configurations.
3. Create web protection HTTP request callback functions.
4. Create web protection policy.
5. Check URL against the cloud.
6. Retrieve the response rating of the URL.
7. Check rating against the local policy to tell if it violates the policy.