Safe Browsing
Quick setup

1.11.2017
Introduction

The Safe Browsing API is a programming interface that lets you check whether a URL is in the lists of suspected malicious pages. The lists are compiled and maintained by Yandex.

This document summarizes and classifies widespread threats. It also describes how to quickly integrate API using freely distributed libraries. The quick setup is available for projects implemented in the following programming languages:

- Python
- PHP
- C#.Net

Overview

As the internet expands, there are a growing number of documents that are potential threats to users' technical devices (computers and mobile devices).

Potentially dangerous documents can be divided into two groups:

- “Malware” causes malicious code to be executed. Running it may cause data to be leaked or lost, and can also harm user devices. It may be authorized (for example, when downloading and running an executable file) or unauthorized (for example, a spyware attack).
- “Phishing” requests confidential user data for further unauthorized use. These pages are copies of sites where the user probably has an account. When the user authenticates, confidential data is intercepted without authorization.

Yandex maintains a database of pages that are considered dangerous for user devices. The database is updated regularly.

External applications have access to the database via the Safe Browsing API (SB API). If the address of a requested document matches a URL from the database, it indicates a high risk of threat posed by this document. Access to the potentially dangerous page is blocked, and the appropriate warning message is returned to the application user.

Access to API

Access to SB API requires an API-key. To get a key, use the feedback form to send a request and tell us the type and volume of data you are planning to check.
Integration

To quickly integrate the SB API with a project implemented in Python, PHP or C#.Net, complete the following steps:

1. Get an API-key.
2. Download library source files.

Libraries are available for projects that are implemented using one of the following programming languages:

<table>
<thead>
<tr>
<th>Programming language</th>
<th>Libraries available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python</td>
<td>SafeBrowsing API Python Client Version 0.3.</td>
</tr>
<tr>
<td>PHP</td>
<td>SafeBrowsing API PHP SDK.</td>
</tr>
<tr>
<td></td>
<td>phpGSB 0.2.3.</td>
</tr>
<tr>
<td>C#.Net</td>
<td>C#.Net</td>
</tr>
</tbody>
</table>

Note:
For C++ projects, you can extract and re-use open code from code bases for the Mozilla Firefox and Chromium browsers.

3. Make corrections to the source code of the selected library.
   The list of changes depends on the library and version you are using:
   - SafeBrowsing API PHP SDK.
   - SafeBrowsing API Python Client Version 0.3.
   - phpGSB v0.2.3.
   - Google Safe Browsing API for .NET (April 25, 2011).

Tip:
Вносить изменения не требуется, библиотека готова к использованию.

Attention!
Libraries are updated regularly. The names of files that contain the required parameters might not match the ones described here if you are using a different version of a library. In this case, you need to follow the general recommendations for changing parameters.

4. Integrate a library function call in your project.
5. Optional. Configure an output page containing detailed information about the suspected document.
   The page is called using an HTTP GET request in the following format:
   
   http://yandex.ru/infected?l10n=<language code for returning the information page (ISO 639–1 format)&url=<address of the infected page>
   

After completing these steps, initialize downloading the SB database using the selected library. Download speed depends on data transfer capacity.
SafeBrowsing API PHP SDK

To download the library, use this link.

The library is ready for use. You do not need to make any changes.

SafeBrowsing API Python Client Version 0.3

To download the library, use this link.

To make changes to a library, follow these steps:

1. Save the diff file in the library directory that contains the client.py.
2. Execute this command:
   ```bash
   patch client.py sb-python.diff
   ```

In some cases (for example, if you are using a different version of the library), it may be necessary to do this manually.

The table below shows a list of source files in the library that require changes.

<table>
<thead>
<tr>
<th>File</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>client.py</td>
<td>1. Find the following section:</td>
</tr>
</tbody>
</table>
|            |   ```python
|            |   def __init__(self, ds, apikey,
|            |   hp=('safebrowsing.clients.google.com', 80),
|            |   ssl_hp=('sb-ssl.google.com', 443), base_path='/safebrowsing',
|            | 2. Substitute the following: |
|            |   ```python
| server.py  | 1. Find the following section: |
|            |   sbls.append(sblist.List(line.strip()))
|            | 2. Substitute the following: |
|            |   if line.strip().endswith('-shavar'):
|            |     sbls.append(sblist.List(line.strip()))

phpGSB v0.2.3

To download the library, use this link.

The table below shows a list of source files in the library that require changes.

<table>
<thead>
<tr>
<th>File</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>phpgsb.class.php</td>
<td>1. Find the following line:</td>
</tr>
</tbody>
</table>
|             |   ```javascript
|             |   var $apiversion = "2.2";
|             | 2. Substitute the following: |
var $apiversion = "2.3";

3. Find the following section:

```php
function googleDownloader($url, $options, $followbackoff=false) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    return curl_exec($ch);
}
```

4. Substitute the following:

```php
function googleDownloader($url, $options, $followbackoff=false) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    return curl_exec($ch);
}
```

5. Find the following line:

```php
$result = $this->googleDownloader("http://safebrowsing.clients.google.com/safebrowsing/downloads?client=api&apikey=".$this->apikey."&appver=".$this->apiversion."&pver=".$this->apiversion.$buildopts."data");
```

6. Substitute the following:

```php
$result = $this->googleDownloader("http://sba.yandex.net/downloads?client=api&apikey=<API key value>"."&appver=2.3"."&pver=".$this->apiversion.$buildopts."data");
```

Attention!
Substitute with the API key value provided to you.

7. Find the following line:

```php
$result = $this->googleDownloader("http://safebrowsing.clients.google.com/safebrowsing/gethash?client=api&apikey=".$this->apikey."&appver=".$this->apiversion.$buildopts."lookup");
```

8. Substitute the following:

```php
$result = $this->googleDownloader("http://sba.yandex.net/gethash?client=api&apikey=<API key value>"."&appver=2.3"."&pver=".$this->apiversion.$buildopts."lookup");
```

Attention!
Substitute with the API key value provided to you.

---

**Google Safe Browsing API for .NET (April 25, 2011)**

To download the library, use this link.

The table below shows a list of source files in the library that require changes.

<table>
<thead>
<tr>
<th>File</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google.SafeBrowsing/API.cs</td>
<td>1. Find the following line:</td>
</tr>
</tbody>
</table>
## General recommendations for changing parameters

Parameters can be stored in configuration files, or they may be hardcoded in the program code.

The way that parameters are stored can differ depending on the version of the library you are using. For this reason, when using a different library version than the ones provided, you must first determine where the parameters to be changed are located.

The list of parameters that need to be changed differs depending on the programming language that the program is written in.

In general, you must find and replace the following lines in the library files, using the data you received when generating an API key:

<table>
<thead>
<tr>
<th>Found string</th>
<th>Modified string</th>
<th>Example</th>
</tr>
</thead>
</table>
Attention!
Substitute with the API key value provided to you.

Processing user complaints

The SB API includes the ability to update the database of potentially harmful documents based on user complaints. When a complaint is received, it is reviewed by Yandex employees. If the threat is confirmed, the URL is added to the list of malicious pages.

User complaints are made using the feedback form for the Yandex.Webmaster service. To embed links in the feedback form, you need to find and modify the following lines in the source files of the library you are using:

<table>
<thead>
<tr>
<th>Found string</th>
<th>Modified string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depending on the library you are using, you need to find one or more of the following lines:</td>
<td></td>
</tr>
<tr>
<td>• <a href="http://www.google.com/safebrowsing/report_badware/">http://www.google.com/safebrowsing/report_badware/</a></td>
<td></td>
</tr>
</tbody>
</table>
Feedback

If you have not found the information you need in the documentation or in the Service Terms of Use, contact SafeBrowsing support service.
Safe Browsing
Quick setup

1.11.2017