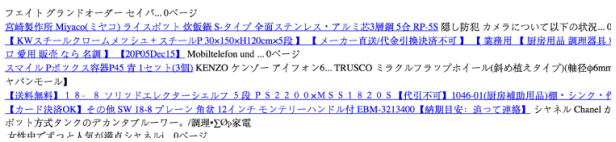
Fixing the Japanese keyword hack

Note: Unsure whether or not your site is hacked? Start by reading our how to check if your site is hacked guide.

This guide is created specifically for a type of hack that creates autogenerated Japanese text to your site which we'll refer to as the Japanese keyword hack. It's designed for users of popular Content Management Systems (CMSs), but you'll find this guide useful even if you don't use a CMS.

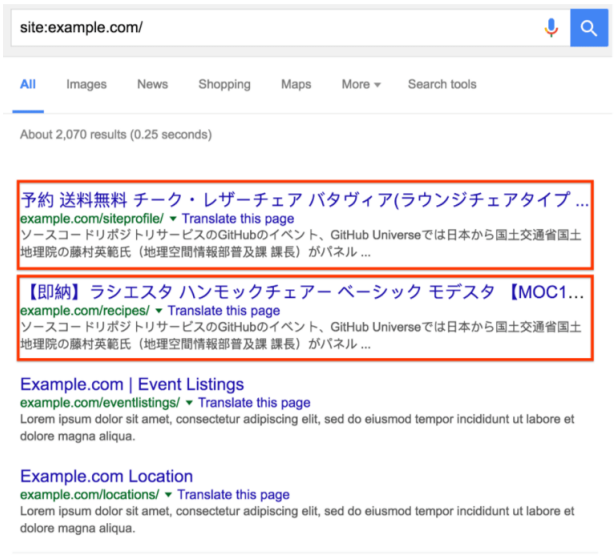
We want to make sure this guide is really helpful to you. Please leave feedback to help us improve!

Identifying this type of hack


フェイト グランドオーダー セイバ... 0ページ
宮崎製作所 Miyaco(ミヤコ) ライスポット 炊飯鍋 S-タイプ 全面ステンレス・アルミ芯3層鋼 5合 RP-5S 隠し防犯 カメラについて以下の状況... 0
【KWスチールクロームメッシュ＋スチールP 30×150×H120cm×5段】【メーカー直送/代金引換決済不可】【業務用【厨房用品 調理器具】
ロ 愛用 販売 なら 名調】【20P05Dec15】 Mobiltelefon und ... 0ページ
スマイルPボックス 容器P45 青 1セット(3個) KENZO ケンゾー アイフォン6... TRUSCO ミラクルフラップホイール(斜め植えタイプ)(軸径φ6mm
ヤパンモール】
【送料無料】１８－８ ソリッドエレクターシェルフ ５段 ＰＳ２２００ｘＭＳＳ１８２０Ｓ【代引不可】1046-01(厨房補助用品)棚・シンク・
【カード決済OK】その他 SW 18-8 プレーン 角盆 12インチ モンテリーハンドル付 EBM-3213400【納期目安：追って連絡】 シャネル Chanel カ
ポット方式タンクのデカンタブルーワー。/調理・ｷｯﾁﾝ家電
女性由ブランド人気が源点シャネル 0ページ

The Japanese keywords hack typically creates new pages with autogenerated Japanese text on your site in randomly generated directory names (for instance, http://example.com/ltjmnjp/341.html). These pages are monetized using affiliate links to stores selling fake brand merchandise and then shown in Google search. Here's an example of what one of these pages look like:

With this type of hack, the hacker will typically add themselves as a property owner in Search Console, in order to increase profits by manipulating your site's settings like geotargeting or sitemaps. If you've received a notification that someone you don't know has verified your site in Search Console, there's a strong possibility that your site has been hacked.



Start by checking the Security Issues tool in Search Console to see if Google has discovered any of these hacked pages on your site. Sometimes you can also uncover pages like this by opening a Google Search window and typing in `site:_your site url_`, with the root level URL of your site. This will show you the pages that Google has indexed for your site, including the hacked pages. Flip through a couple of pages of search results to see if you spot any unusual URLs. If you don't see any hacked content in Google Search, use the same search terms with a different search engine. An example of what that would look like is below.

Note: Notice that the search results here contain many pages not created by the site owner. If you look closely at the descriptions, you'll see examples of the gibberish text that this hack creates.

Typically, when you click a link to a hacked page you will either be redirected to another site, or you will see a page full of gibberish content. However, you might also see a message suggesting that the page does not exist (for example, a 404 error). Don't be fooled! Hackers will try to trick you into thinking the page is gone or fixed when it's still hacked. They do this by cloaking content. Check for cloaking by entering your site's URLs in the Fetch as Google tool. The Fetch as Google tool allows you to see the underlying hidden content.

If you see these issues, your site has most likely been affected by this type of hack.

Fixing the hack

Before you start, make an offline copy of any files before you remove them, in case you need to restore them later. Better yet, back up your entire site before you start the cleanup process.

You can do this by saving all the files that are on your server to a location off your server or searching for the best backup options for your particular Content Management System (CMS). If you're using a CMS, you should back up the database as well.

Remove newly created accounts from Search Console

If a new owner you don't recognize has been added to your Search Console account, revoke their access as soon as possible. You can check which users are verified for your site on the Search Console verification page. Click on "Verification Details" for the site to view all verified users.

To remove an owner from Search Console, read through the Remove Owner section of the Managing users, owners, and permissions Help Center. You'll need to remove the associated verification token which is typically either an HTML file on the root of your site or a dynamically generated **.htaccess** file mimicking an HTML file.

If you are unable to find an HTML verification token on your site, check for a rewrite rule in your **.htaccess** file. The rewrite rule will look similar to this:

```
RewriteEngine On
  RewriteRule ^google(.*)\.html$ dir/file.php?google=$1 [L]
```

Note: You can typically check if you've successfully removed a dynamically generated verification token by navigating to a simulated verification token file like **`wwww.example.com/google[random number and letters].html`**. For example, if your site is www.brandonsbaseballcards.com, try navigating to www.brandonsbaseballcards.com/google1234.html. If that page returns a HTTP 404, then the dynamically generated verification token is likely fixed.

To remove the dynamically generated verification token from your **.htaccess** file, follow the steps below.

Check your **.htaccess** file (2 steps)

Aside from using a **.htaccess** file to create dynamically generated verification tokens, hackers often use **.htaccess** rules to redirect users or create gibberish spammy pages. Unless you have custom **.htaccess** rules, consider replacing your **.htaccess** with a completely new copy.

Step 1

Locate your `.htaccess` file on your site. If you're not sure where to find it and you're using a CMS like WordPress, Joomla, or Drupal, search for ".htaccess file location" in a search engine along with the name of your CMS. Depending on your site, you might see multiple `.htaccess` files. Make a list of all of `.htaccess` file locations.

Note: The `.htaccess` is often a "hidden file." Make sure to enable showing hidden files when you're searching for it.

Step 2

Replace all `.htaccess` files with a clean or default version of the `.htaccess` file. You can usually find a default version of a `.htaccess` file by searching for "default `.htaccess` file" and the name of your CMS. For sites with multiple `.htaccess` files, find a clean version of each one and perform the replacement.

If no default `.htaccess` exists and you've never configured an `.htaccess` file on your site, the `.htaccess` file you find on your site is probably malicious. Save a copy of the `.htaccess` file(s) offline just in case and delete the `.htaccess` file from your site.

Remove All Malicious Files and Scripts (4 steps)

Identifying malicious files can be tricky and can take several hours. Take your time when checking your files. If you haven't yet, this is a good time to back up the files on your site. Do a Google search for "back up site" and the name of your CMS to find instructions on how to back up your site.

Step 1

If you use a CMS, reinstall all the core (default) files that come in the default distribution of your CMS, as well as anything you may have added (such as themes, modules, plugins). This helps ensure that these files are clear of hacked content. You can do a Google search for "reinstall" and your CMS name to find instructions on the reinstallation process. If you have any plugins, modules, extensions, or themes, make sure to reinstall those as well.

Caution: Reinstalling your core files can cause you to lose any customizations that you've made. Be sure to create a backup of your database and all files before you reinstall.

Step 2

Hackers will often modify your sitemap or add a new sitemaps to help get their URLs indexed more quickly. If you previously had a sitemap file, check the file for any suspicious links and remove them from your sitemap. If there are any sitemap files you don't remember adding to your site, double check them and remove them if they only contain spammy URLs.

Step 3

Look for any other malicious or compromised files left. You might have already removed all malicious files in the previous two steps, but it's best to work through these next few steps in case there are more files on your site that have been compromised.

Don't get overwhelmed by thinking that you need to open and look through every PHP file. Start by creating a list of suspicious PHP files that you want to investigate. Here are a few ways to determine which PHP files are suspicious:

- If you've already reloaded your CMS files, look only at files that are not part of your default CMS files or folders. This should rule out a large number of PHP files and leave you with a handful of files to look at.
- Sort the files on your site by the date they were last modified. Look for files that were modified within a few months of the time that you first discovered your site was hacked.
- Sort the files on your site by size. Look for any unusually large files.

Note: Attackers commonly inject scripts into the following files: `index.php`, `wp-load.php`, `404.php`, and `view.php`.

Step 4

Once you have a list of suspicious PHP files, check to see if they are malicious. If you're unfamiliar with PHP, this process might be more time consuming, so consider brushing up on some PHP documentation. If you're completely new to coding, we recommend getting help. In the meantime, there are some basic patterns that you can look for to identify malicious files.

If you use a CMS, and are not in the habit of editing those files directly, compare the files on your server to a list of the default files packaged with the CMS and any plugins and themes. Look for files that do not belong, as well as files whose sizes seem larger than their defaults.

Scan through the suspicious files you've already identified to look for blocks of obfuscated code. This might look like a combination of seemingly jumbled letters and numbers. The obfuscated code is usually preceded by a combination of PHP functions like **base64_decode**, **rot13**, **eval**, **strrev**, **gzinflate**. Here is an example of what the block of code might look like.

Sometimes all this code will be stuffed into one long line of text, making it look smaller than it actually is.

```
$O_OOO_OO_0=urldecode("%6E1%7A%62%2F%6D%615%5C%76%740%6928%2D%70
%78%75%71%79%2A6%6C%72%6B%64%679%5F%65%68%63%73%77%6F4%2B%6637%6A");
$OOO_OOOO__=$O_OOO_OO_0{26}.$O_OOO_OO_0{6}.$O_OOO_OO_0{10}.$O_OOO_OO_0{30}
```

Check to see if your site is clean

Once you're done getting rid of hacked files, check to see if your hard work paid off. Remember those gibberish pages you identified earlier? Use the Fetch as Google tool on them again to see if they still exist. If they respond as "Not Found" in Fetch as Google, chances are you're in pretty good shape and you can move on to fixing the vulnerabilities on your site.

Note: You can also follow the steps in the Hacked Sites Troubleshooter to check if there's still hacked content on your site.

How do I prevent getting hacked again?

Fixing vulnerabilities on your site is an essential final step for fixing your site. A recent study done found that 20% of hacked sites get hacked again within 1 day. Knowing exactly how your site was hacked is helpful. Read our top ways websites get hacked by spammers guide to start your investigation. However, if you're unable to find out how your site was hacked, below is a checklist of things you can do reduce vulnerabilities on your site.

- **Regularly scan your computer:** Use any popular virus scanner to check for viruses or vulnerabilities.
- **Regularly change your passwords:** Regularly changing the passwords to all your website accounts like your hosting provider, FTP, and CMS can prevent unauthorized access to your site. It's important to create a strong, unique password for each account.
- **Use Two-Factor Authentication (2FA):** Consider enabling 2FA on any service that requires you to log in. 2FA makes it harder for hackers to log in even if they successfully steal your password.
- **Update your CMS, plugins, extensions, and modules regularly:** Hopefully you've already done this step. Many sites get hacked because of the outdated software running on a site. Some CMSs support auto-updating.

- **Consider subscribing to a security service to monitor your site:** There's a lot of great services out there that can help you monitor your site for a small fee. Consider registering with them to keep your site safe.

Additional resources

If you're still having trouble fixing your site, there are a few more resources that might help you.

These tools scan your site and may be able to find problematic content. Other than VirusTotal, Google doesn't run or support them.

Virus Total, Aw-snap.info, Sucuri Site Check, Quttera: These are just some tools that may be able to scan your site for problematic content. Keep in mind that these scanners can't guarantee that they will identify every type of problematic content.

Here are additional resources from Google that can help you:

- Google Webmaster Help Forum
- Google Webmaster Help for Hacked Sites
- Google SafeBrowsing

Note: Missing a tool you think might be useful? Leave feedback and let us know.

Last updated July 20, 2018.