Fixing the gibberish 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 adds keyword-heavy gibberish pages to your site which we'll refer to as the gibberish 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!

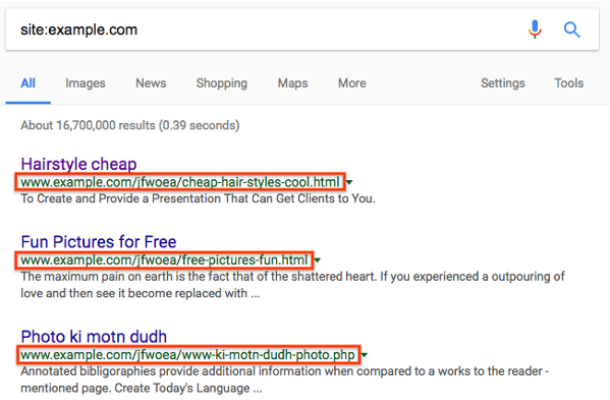
Identifying this type of hack

The gibberish hack automatically creates many pages with non-sensical sentences filled with keywords on your site. These are pages that you didn't create, but have URLs that might be compelling for users to click. Hackers do this so the hacked pages show up in Google Search. Then, if people try to visit these pages, they'll be redirected to an unrelated page, like a porn site. Hackers make money when people visit these unrelated pages. Here are some examples of the type of files you might see on a site affected by the gibberish hack:

- www.example.com/cheap-hair-styles-cool.html
- www.example.com/free-pictures-fun.html
- www.example.com/nice-song-download-file.php

Sometimes they appear in a folder composed of random characters and use different languages:

- www.example.com/jfwoea/cheap-hair-styles-cool.html
- www.example.com/jfwoea/free-pictures-fun.html
- www.example.com/jfwoea/www-ki-motn-dudh-photo.php
- www.example.com/jfwoea/foto-cewe-zaman-sekarang.php



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.

Check your `.htaccess` file (2 steps)

The gibberish hack redirects visitors from your site using the `.htaccess` file.

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.

Finding and removing other malicious files (5 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

Now you need to look for any other malicious or compromised files left. This is the most difficult and time-consuming part of the process, but after this you're almost done!

This hack typically leaves two types of files: `.txt` files and .php files. The `.txt` files serve are template files, and the `.php` files determine what type of non-sensical content to load onto your site. Start by looking for the `.txt` files. Depending on how you're connecting to your site, you should see some type of search functionality for files Search for ".txt" to pull up all the files with a `.txt` extension. Most of these will be legitimate files of text like license agreements, readme files, and so on. You're looking for a particular set of `.txt` files that contain HTML code used to create spammy templates. Below are snippets of different pieces of code that you might find in these malicious `.txt` files.

Hackers use keyword replacement to create the spammy pages. You'll most likely see some type of generic word that can be replaced throughout the hacked file.

```
<title>{keyword}</title>
<meta name="description" content="{keyword}" />
<meta name="keywords" content="{keyword}" />
<meta property="og:title" content="{keyword}" />
```

Additionally, most of these files contain some type of code that positions spammy links and spammy text off the visible page.

```
<div style="position: absolute; top: -1000px; left: -1000px;">
  Cheap prescription drugs
</div>
```

Remove these `.txt` files. If they're all in the same folder you can remove the entire folder.

Step 3

The malicious PHP files are a bit harder to track down. There could be one or many malicious PHP files on your site. They could all be contained in the same subdirectory or scattered around your site.

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:

- Since you’ve already reloaded your CMS files, look only at files that are not part of your default CMS files or folders. This should eliminate a large number of PHP files and leave you with a handful of files to look at.
- Sort the files on your site by last modified date. 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.

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.

First, scan through the suspicious files you’ve already identified to look for large blocks of text with a combination of seemingly jumbled letters and numbers. The large block of text 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.

```
// Hackers try to confuse webmasters by encoding malicious code into
// blocks of texts. Be wary of unfamiliar code blocks like this.

base64_decode(strrev("hMXZpRXaslmYhJXZuxWd2BSZ0l2cgknbhByZul2czVmckRWYgknYgM3ajFG
ZgknbhBSbvJnZgUGdpNHIyV3b5BSZyV3YlNHIvRHI0V2Zy9mZgQ3Ju9GRg4SZ0l2cgIXdvlHI4lmZg4WY
hVmcnBydvJGblBiZvBCdpJGIhBCZuFGIl1Wa0BCa0l2dgQXdCBiLkJXYoBSZiBibhNGIlR2bjBycphGdg
lRGI5xWZ0Fmb1RncvZmbVBiLn5WauVGcwFGagM3J0FGa3BCZuFGdzJXZk5Wdg8GdgU3b5BicvZGI0xWdj
1GIvRHIzlGa0BSZrlGbgUGZvNGIlRWaoByb0BSZrlGbgMnclt2YhhEIuUGZvNGIlxmYhRWYlJnb1BychB
1GIlR2bjBCZlRXYjNXdmJ2bgMXdvl2YpxWYtBiZvBSZjVWawBSYgMXagMXaoRFIskGS"));
```

Sometimes the code isn’t jumbled and just looks like normal script. If you’re not certain whether or not the code is bad, stop by our Webmaster Help Forums where a group of

experienced webmasters can help you look over the files.

Step 5

Now that you know which files are suspicious, create a backup or a local copy by saving them onto your computer just in case it wasn't malicious, and delete the suspicious files.

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.