Assess malware damage

This step pertains to sites hacked to distribute malware, often with the warning *"This site may harm your computer"* displayed in search results. It's one of the longest steps in the recovery process. In this step, you'll compile a list of the damaged files on your site. You'll use this list in a later step, Clean and maintain your site.

Note: 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).

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.

What you'll do

This step has three sections:

- Preparation
- Investigation of specific malware infection types
- General filesystem damage assessment

Preparation

1. 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.
2. 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.
3. Check out additional resources as necessary:
4. Watch the video above to see how malware works and how you can stay safe during your malware investigation.
5. View the Google Safe Browsing diagnostic page to see public information about whether a site is potentially harmful to users. You can see the listing status for your site at a URL similar to the following:

   ```
   http://www.google.com/safebrowsing/diagnostic?site=<your-site>
   ```

   For example: **`http://www.google.com/safebrowsing/diagnostic?site=webmastercentralblog.blogspot.com`** 4. 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>
   ```

Investigation

Investigation of specific malware infection types on your site

1. Select your verified Site in Search Console, then click **Security Issues**.
2. Investigate all malware categories (e.g., Server configuration, SQL injection) listed in Security Issues for your site. Additional information about infected URLs from the category can be found by clicking "Show details." (Details may include sample code snippets injected by the hacker.) For each category, copy into your investigation document:
   - All example infected URLs listed for the malware category in Security Issues.
   - Any additional damaged pages you uncover during your investigation.

- Detailed fIndings about the infected URLs, such as the type of damage caused.

3. Information to aid in the investigation for each malware type is below:
    - Server configuration
    - SQL injection
    - Error template
    - Code injection

Filesystem damage assessment

Next, you'll need to log in to your site's filesystem for more in-depth investigation. Be aware that -- among other things -- 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 "backdoors" that will allow the hacker re-entry to your site or that will continue performing malicious tasks if not deleted.

If your site is online, you can take it back offline for this step.

1. If you have a known good backup of your site, determine which files have been created or modified since the backup. Add these files to your list since you'll likely want to investigate further. On Unix-based systems, you can use a command such as:

    ```
    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
```

2. 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.)
3. Check **configuration files**, such as `.htaccess` and `httpd.conf`, for redirects. Hackers often create conditional redirects based on the user-agent, time of day, or referrer.
4. Check for too 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 {} \;
   ```

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

Previous
← Overview

Next
Identify the vulnerability →

Last updated January 8, 2018.