Clean and maintain your site

You'll need

- Shell/terminal administrator access to your site's servers: web, database, files
- Knowledge of shell/terminal commands
- Understanding of code (such as PHP or JavaScript)
- Storage to create backups of your site (including the files, database, images, etc.)

Next actions

We'll cover several actions in this step:

- Where to locate additional resources if you believe the hacker intended to obtain users' personal information (such as with phishing pages)
- Option to use Remove URLs in Search Console to expedite removal of entirely new, undesirable, user-visible URLs created by the hacker that you don't want surfacing in Google Search results.
- Option to use Fetch as Google in Search Console to expedite processing by Google of clean pages – meaning pages that are either new or newly updated – that you want to appear in Google search results
- Installation of the latest, most secure version of software
- Removal of unnecessary or unused applications or plugins that could make your site more vulnerable in the future
- Restoring good content and eliminating the hacker's content
- Fixing the root cause vulnerability exploited by the hacker

- Changing all passwords
- Planning to keep your site secure

1. Locate support resources

Locate support resources when dealing with loss of confidential information, such as with phishing pages.

If confidential user information was obtained from your site (for example, because it was part of a phishing attack), you may want to consider any business, regulatory, or legal responsibilities before you begin cleaning your site or deleting any files. In cases of phishing, antiphishing.org has useful resources such as their document What to do if you your site has been hacked by phishers.

2. Consider expediting removal of new URLs created by the hacker

If the hacker created entirely new, user-visible URLs, you can have these pages more quickly removed from Google search results by using the Remove URLs feature in Search Console. This an entirely optional step. If you simply delete the pages and then configure your server to return a 404 status code, the pages will naturally fall out of Google's index with time.

- The decision to use URL Removal will likely depend on the number of new, unwanted pages created (too many pages might be cumbersome to include in Remove URLs), as well as the potential damage these pages could cause users. To keep the pages submitted through URL Removal from ever appearing in search results, be sure the pages are also configured to return a 404 File not Found response for the unwanted/removed URLs.
- Don't use this tool to request removal of any previously good pages that were only damaged by the hacker – you'll want these pages to appear in search results once they're cleaned up. *URL Removal is only for pages you'll never want to appear in results.*

3. Consider expediting processing by Google of your clean pages

If you have new or updated clean pages, you can use the Fetch as Google feature in Search Console to submit these pages to Google's index. This is entirely optional; if you skip this step, your new or modified pages will likely be crawled and processed with time.

4. Start cleaning up your server(s)

Now is the time to start cleaning up your site based on the notes you took during Assess the damage and Identify the vulnerability. The following path you'll take in this step depends on the type of backup you have available.

- Clean and current backup
- Clean but outdated backup
- No backup available

First, check that this backup was created **before** your site was hacked.

Clean and current backup

1. Restore your backup.
2. Install any software upgrades, updates, or patches available. This includes software for the OS if you're in control of the server, and all applications, such as the content management system, ecommerce platform, plugins, templates, etc.
3. Consider whether you can eliminate software on your server (e.g., widgets, plugins, or applications) that the site no longer uses.
4. Correct the vulnerability.
5. Make sure all issues found during Assess the damage are addressed.
6. Change the passwords one more time for all accounts related to the site (e.g., logins for FTP access, database access, system administrators, and CMS accounts). On Unix-based systems:

   ```
   passwd admin1
   ```

Clean but outdated backup

1. Make a disk image of your current site even though it's still infected. This copy is just for safety. Mark the copy as infected to distinguish it from the others. On a Unix-based system, making a disk image could be:

   ```
   dd if=/dev/sda bs=1024 conv=noerror,sync | gzip -c -9 \
     > /mirror/full-backup-20120125-infected.gz
   ```

2. Make a backup file system copy of your server, including images and media files. If you have a database, backup the database as well.

```
tar -pczf full-backup-20120125-infected.tar.gz www/
mysqldump -u root -p --all-databases | gzip -9 \
  > fulldb_backup-20120125-infected.sql
```

3. Restore the clean but outdated backup on your server.
4. Consider whether you can eliminate software on your server (e.g., widgets, plugins, or applications) that the site no longer uses.
5. Upgrade all software, including the OS if you're in control of the server, and all software applications, such as the content management system, ecommerce platform, plugins, templates, etc. Be sure to check and install available security updates and patches.
6. Correct the vulnerability.
7. Perform a site **`diff`** either manually or in an automated fashion – between the clean backup and the current infected copy.

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

8. Upload any new, clean content that you'd like to preserve from the infected copy on the upgraded server.

```
rsync -avz /backups/full-backup-20120124/www/clean-file.jpg /www/
```

9. Check that each URL listed from Assess the damage has been corrected.
10. Change the passwords one more time for all accounts related to the site (for example, logins for FTP access, database access, system administrators, and CMS accounts). On Unix-based systems:

```
$passwd admin1
```

No backup available

1. Make two backups of your site even though it's still infected. Having an extra backup will help recover accidentally deleted content, or allow you to revert and try again if things go awry. Label each backup with "infected" for future reference.
    - One of your backups will be a disk image or "clone version" of your site. This format makes restoring content even simpler. You can leave the disk image aside for an emergency. On a Unix-based system, making a disk image could be:

```
dd if=/dev/sda bs=1024 conv=noerror,sync | gzip -c -9 \
  > /mirror/full-backup-20120125-infected.gz
```

- The other backup will be a file system copy from your server, including images and media files. If you have a database, backup the database as well.

```
tar -pczf full-backup-20120125-infected.tar.gz www/
mysqldump -u root -p --all-databases | gzip -9 \
  > fulldb_backup-20120125-infected.sql
```

- If you don't have a disk image, make two backups of the database and two backups of the filesystem.

2. Clean the site's content on the new backup filesystem copy (not the server itself)
    a. If your earlier investigation found too lenient file permissions, go ahead and correct them. Make sure you do this on the backup copy, not the server itself.
    b. Also on the backup copy, clean all files corresponding to the URLs discovered as compromised from Assess the damage. These may be server configuration files, JavaScript, HTML, PHP.
    c. Make sure to also remove (serve a 404 response) for new files created by the hacker (which you may or may not have submitted using the URL Removal tool in Search Console.)
    d. Correct the vulnerability if it exists in your code or cracked passwords. Input validation libraries or security audits may be of help.
    e. If your site has a database, begin cleaning up hacker-modified records in your backup. Just before you think you're done, perform a sanity check on more of the records to make sure it looks clean.
    f. Change the passwords one more time for all accounts related to the site (e.g., logins for FTP access, database access, system administrators, and CMS accounts). On Unix-based systems:

    $passwd admin1

    g. At this point, the once-infected backup copy of your site should contain only clean data. Keep this clean copy on the side and move to action #5.

5. Eliminate unnecessary software

Consider whether you can eliminate software on your server – such as widgets, plugins, or applications – that the site no longer uses. This can increase security and simplify future

maintenance.

6. Clean all servers

1. Perform a clean installation, not just an upgrade. Upgrades may leave files from a previous version. If an infected file remains on the server, another hack is more likely to occur.
    - The fresh installation should include the OS if you're in control of the server, and all software applications, such as the content management system, ecommerce platform, plugins, templates, etc. Be sure to check for available security updates and patches.
2. Transfer good content from the clean backup filesystem copy to the freshly installed server(s). Upload/restore only the known clean files/database. Be sure to maintain the appropriate file permissions and not to overwrite the freshly installed system files.
3. Perform one last change of passwords for all accounts related to the site (e.g., logins for FTP access, database access, system administrators, and CMS accounts). On Unix-based systems:

    ```
    passwd admin1
    ```

7. Create a long-term maintenance plan

There are many informative resources on the web for strong site maintenance, like StopBadware's Preventing badware: basics. We also strongly recommend that you do the following:

- Make regular, automated backups of your site.
- Be vigilant about keeping software updated.
- Understand the security practices of all applications, plugins, third-party software, etc., before you install them on your server. A security vulnerability in one software application can affect the safety of your entire site.
- Enforce creation of strong passwords.
- Keep all devices used to log in to the machine secure (updated operating system and browser).

8. Double-check that cleanup is complete

Make sure you can answer "yes" to the following questions:

- Have I taken the proper steps if the hacker obtained users' personal information?
- Is my site running the latest, most secure version of software?
- Have I removed all unnecessary or unused applications or plugins that could make my site more vulnerable in the future?
- Did I restore my content and eliminate the hacker's content?
- Did I fix the root cause vulnerability that allowed my site to be hacked?
- Do I have a plan to keep my site secure?

9. Bring your site back online

Previous
← Identify the vulnerability

Next
Request a review →

Last updated January 8, 2018.