

Forum:[Hack Forums](#)

Thread: [Google Safe Browsing | Explanation and reasons | How to get rid of red warning screen](#)

Ph4n70M Phr34K - June 23, 2023, 9:58 am

Related threads:

<https://hackforums.net/showthread.php?tid=6243986>

<https://hackforums.net/showthread.php?tid=6243938>

<https://hackforums.net/showthread.php?tid=6233658>

<https://hackforums.net/showthread.php?tid=6244180>

What is Google Safe Browsing?

Google Safe Browsing is a security service provided by Google that protects users against malicious webpages. This service checks URLs against Google's constantly updated lists of unsafe websites. These could include websites that are known to host malware or phishing attempts.

When you are about to visit a website suspected of these kinds of security risks, Google Safe Browsing will show a warning message indicating that the site you're about to visit may be dangerous. This gives users an opportunity to avoid that webpage and keep their devices and personal information secure.

Google Safe Browsing is utilized by majority of browsers. Additionally, Google provides APIs that allow other software developers to use this service in their applications or services to enhance user security.

Google Safe Browsing consists of four parts:

- phishing detection mechanism built into chrome based on matching color palettes and parts of domains of previously visited sites.
- a mechanism for checking url-addresses and hashes of files for a match against a preloaded database of shortened hashes stored locally.
- a mechanism for unpacking most types of archives, and checking hashes of executable files and hashes of their sections against a locally stored database.
- a crawler that receives the addresses of downloaded files, goes through them, downloads them and transfers them to Google servers for analysis by the anti-virus engine using the VirusTotal service.

Reasons why a website could be flagged as dangerous.

Google Safe Browsing can flag a website for a variety of reasons, all related to the safety and security of users. Some common reasons include:

- Phishing: The website may be suspected of phishing, or attempting to trick users into revealing personal information such as usernames, passwords, or credit card numbers.
- Malware: The website may be suspected of hosting malware, or malicious software that can harm users' devices or steal information.
- Unwanted Software: The site may offer downloads that violate Google's Unwanted Software Policy, such as software that is deceptive, affects the user's system in unexpected ways, doesn't do what it promises, or is bundled with other software in a confusing way. This also includes applications that are malware, phishing, or that carry out other harmful behaviors like unauthorized data collection.
- Social Engineering (Deceptive Content): Websites that trick users into doing something dangerous, for example, by telling them their software is out-of-date (like a fake flash update), or their device is infected with a virus.

If your website is flagged by Google Safe Browsing, you can use Google Search Console to find out why it was flagged and resolve the issue. You can then request a review of your website from Google, and if the issue has been resolved, the warning will be removed.

Phishing detection based on matching color palettes and domain parts of previously visited websites.

Chrome counts and stores the sum of the pixels of each color on all the pages you visit. Thus, if you copy the design of a login form or landing page one-to-one, it will be identified as phishing. To confuse this mechanism, it is enough to use the offset of the color palette through CSS.

Here is an example of JavaScript code that should be placed in the head section of the site, inside the "script" tag:

```
Code document.documentElement.style.cssText="filter:hue-rotate(4deg)";
```

Chrome also checks that previously visited domains and sub-domains and their parts match those that it sees in the address bar.

Thus, domains like telegram.blabla.com or att.net.fuckyou.com or ziraatbank-login.oh-my-site.com will be identified as phishing if the user has visited these sites before. The solution is to use typos in words, or not to use branded words at all, using common ones, for example, sso-login.zirat-office.com.

Chrome sharpens the mechanisms for checking for phishing if your phish contains an `<input type="password">` password input field or a file is downloaded. In order to get around this protection, you can use an alternative to the input tag and use the `contenteditable="true"` attribute on any HTML tag.

An example of the implementation of turning any tag into a text input can be found here:
<http://jsfiddle.net/Onojd66r/>

Mechanism for checking URL-addresses and hashes of files for a match against a preloaded database of shortened hashes stored locally.

Just don't use "dirty" domains. In the context of Google Safe Browsing, a "dirty" domain refers to the domain name of a website that has been identified as potentially harmful. However, the domain

name itself does not necessarily give any indication that the website is harmful.

When a website is identified as harmful by Google Safe Browsing, it's usually due to malicious content or behavior found on that site, not the actual domain name itself. That said, there can sometimes be signs in the domain name that may suggest it's suspicious, such as:

- Gibberish or Nonsense Domain Names: Cybercriminals sometimes use random, computer-generated strings of letters and numbers to register domain names. These can be a sign of a malicious site.
- Domain Names Mimicking Well-Known Brands: This is a common tactic used in phishing attempts. The domain name may include the name of a well-known brand, but with small variations in spelling or with additional words or characters. For example, "bankofarnerica.com" instead of "bankofamerica.com".
- Domains with Many Hyphens or Subdomains: Legitimate domains usually have a relatively simple structure, while malicious domains may use many hyphens or multiple subdomains to trick users. For example, "www.bank-of-america-login.com" or "www.login.bankofamerica.security.com".
- Domains Using Less Common Top-Level Domains (TLDs): While many legitimate sites use less common TLDs (like .xyz, .info, etc), a high percentage of malicious sites also use these to take advantage of their low cost and availability.

However, these are only potential indicators and not definite proof that a domain is "dirty" or harmful. A domain's reputation in Google Safe Browsing ultimately depends on the content and behavior of the site associated with that domain, not the domain name itself. And even a normally safe website could be compromised and become harmful without the domain name changing.

If a website is deemed to be malicious, Google will flag it in their Safe Browsing database, and then when users attempt to navigate to it using a browser that utilizes Google Safe Browsing (like Chrome or Firefox), they will be shown a warning page. This is to protect users from visiting websites that could harm their devices or steal their personal information.

What if domain has been already burned down.

Go to https://search.google.com/search-console/welcome?utm_source=about-page

We add our domain as a "domain resource" - the first option of two. We confirm the ownership of the domain through a record in the DNS, and on the page that appears you will see the reason for the reddening of your domain. Having eliminated it (if possible, for example, by deleting a phishing page, or a file recognized as malicious), apply for a recheck and wait. Within an hour or two, your domain will no longer be detected as malicious.

Checking hashes of executable files and hashes of their sections against a locally stored database.

Use a good crypt, obfuscate landing js files if they contain code related to cloaking and phishing. Putting the file in the archive will not help - this only makes sense if you're bypassing Chrome's total ban on downloading certain file types, such as .lnk.

Fighting with Google Safe Browsing crawler.

The google ads verification system and the GoogleSafeBrowsing crawler use the same chrome

engine with a number of modifications. Including, the chromium instance is controlled via the webdriver protocol. But half a year ago, chrome fixed the last known way to detect this connection.

Filtering by ip does not make sense, because the crawler uses proxies, including up to Kazakhstan. Thus, the general advice would be to block all countries that do not interest you at the TDS level. DNS requests will still come from google subnets. However, at the moment, filtering by the names of subnets and organizations at the DNS level is not implemented in any known DNS server.

All addresses of files downloaded by users are sent to Google for their subsequent pumping by the crawler. To get around this, you can use the creation of temporary blob-urls via javascript, or binding each url to the user's ip at the server level (php), encoding the hash of the ip-address in the link, and giving the file only if it matches.

Here is an example of creating dynamic urls and attaching them to a download button:

Code < script >

```
document.querySelectorAll('.download').forEach(function(a){
  a.onclick =function( ) {
    fetch('https://url-to-download-file').then(respond = > respond.blob( ) ).then(function(blob, url, a)
{
  a = document.createElement(
'a');
  a.style.display ='none';
  a.href = window.URL.createObjectURL(blob);
  a.download ='desired-filename.exe';
  document.body.appendChild(a);
  a.click( );})returnfalse;}} )< / script >< styletype=
```

```
"text / css">.download {cursor:pointer;}< / style >
```

This code will put a download handler on all buttons that have the "download" class in their styles

Chrome blocking downloads of suspicious file.

Chrome blocks downloading a file from an unfamiliar domain in four cases:

- file download not triggered by user click (sending a .click() event via javascript won't help)
- the file is downloaded from a domain that the user has not visited in the last 24 hours.
- the file is executable, or the archive containing the file contains executable files.
- the file is not signed.

This mechanism is described here:

https://source.chromium.org/chromium/chromium/src/+main:chrome/browser/resources/safe_browsing/README.md;l=143;drc=591c5c478112625a5da995afeeb6566429c04ef9

The list of files that are considered executable can be found in the chromium sources:

https://source.chromium.org/chromium/chromium/src/+main:components/safe_browsing/core/resources/download_file_types.asciipb;l=2172;drc=af17ad3f07c1d8a24381eb7669bec0c2ffb86521

Chrome does not validate the file signature, because it is demanding on network traffic. Therefore, it is important to attach to the file any signature stolen from any other white file or installer, using the SigThief utility, for example. It's free and available on GitHub:

<https://github.com/secretsquirrel/SigThief>

Chrome has a list of trusted domains that I don't know for sure. However, googleDrive and Microsoft OneDrive services definitely belong to it.

By uploading a file to OneDrive and making it "available to everyone", you can get a direct link to download the file through online services, for example

<https://metadataconsulting.blogspot.com/2019/03/OneDrive-2019-Direct-File-Download-URL-Maker.html>

This will allow you to serve the file directly, as long as it is signed. This url can also be wrapped with a url shortener like <https://t.ly/>

Also, the file name should not contain branded words that were contained in the names of popular and previously known files, for example, the words telegram, firefox and the like. Use off-topic or generic words such as Installer, homonyms, or numbers.

Domain Age.

Domain age is one of the decisive factors in the scoring system of the google safe browsing heuristic. Don't use new domains - they will burn like matches. Either use the domain shadowing technique, or use domains with a hold of at least 7 days.

Final words.

This time the thread wasn't written by me. I have only translated it from Russian to English and added some things for better understanding the problem and context. The translation may not be perfect so excuse me for that.

All credits goes to "Aels": /profile/31147-aels/

Original thread: /topic/218301/

Thank you for reading, please leave a QL or HQ post if you enjoyed it.
Stay tuned for more content.
Sincerely, LEGENDS

Monyito666 - June 23, 2023, 11:07 am

thanks for this article :)

Ph4n70M Phr34K - June 26, 2023, 4:31 pm

(Jun 23, 2023 11:07 AM)Monyito666 Wrote: [\[CITE\]](#) thanks for this article :)

No problem, stay tuned for more high quality threads.

Pol.ar - June 26, 2023, 8:12 pm

Didn't expect google safe browsing to honestly be that sophisticated. Great thread man :)

Ph4n70M Phr34K - June 27, 2023, 4:25 pm

(Jun 26, 2023 08:12 PM)Pol.ar Wrote: [\[CITE\]](#) Didn't expect google safe browsing to honestly be that sophisticated. Great thread man :)

Thanks. To be honest i had similar thoughts on this topic, only discovered how it is recently when started digging.

Real kudos to that Russian who wrote it.

katia van dees - July 23, 2023, 7:35 pm

Thanks for sharing this.

Ph4n70M Phr34K - July 25, 2023, 11:31 am

(Jun 29, 2023 08:35 AM)ZELOS Wrote: [\[CITE\]](#) [Ph4n70M Phr34K](#) is an animal!

Thank you for adding such awesome content to HF brother. ??

Nah, im just a transparent squire :)

loadman - July 31, 2023, 3:31 am

(Jun 23, 2023 09:58 AM)Ph4n70M Phr34K Wrote: [\[CITE\]](#)

THE FALLEN

Related threads:

<https://hackforums.net/showthread.php?tid=6243986>

<https://hackforums.net/showthread.php?tid=6243938>

<https://hackforums.net/showthread.php?tid=6233658>

<https://hackforums.net/showthread.php?tid=6244180>

What is Google Safe Browsing?

Google Safe Browsing is a security service provided by Google that protects users against malicious webpages. This service checks URLs against Google's constantly updated lists of unsafe websites. These could include websites that are known to host malware or phishing attempts.

When you are about to visit a website suspected of these kinds of security risks, Google Safe Browsing will show a warning message indicating that the site you're about to visit may be dangerous. This gives users an opportunity to avoid that webpage and keep their devices and personal information secure.

Google Safe Browsing is utilized by majority of browsers. Additionally, Google provides APIs that allow other software developers to use this service in their applications or services to enhance user security.

Google Safe Browsing consists of four parts:

- phishing detection mechanism built into chrome based on matching color palettes and parts of domains of previously visited sites.
- a mechanism for checking url-addresses and hashes of files for a match against a preloaded database of shortened hashes stored locally.
- a mechanism for unpacking most types of archives, and checking hashes of executable files and hashes of their sections against a locally stored database.
- a crawler that receives the addresses of downloaded files, goes through them, downloads them

and transfers them to Google servers for analysis by the anti-virus engine using the VirusTotal service.

Reasons why a website could be flagged as dangerous.

Google Safe Browsing can flag a website for a variety of reasons, all related to the safety and security of users. Some common reasons include:

- Phishing: The website may be suspected of phishing, or attempting to trick users into revealing personal information such as usernames, passwords, or credit card numbers.
- Malware: The website may be suspected of hosting malware, or malicious software that can harm users' devices or steal information.
- Unwanted Software: The site may offer downloads that violate Google's Unwanted Software Policy, such as software that is deceptive, affects the user's system in unexpected ways, doesn't do what it promises, or is bundled with other software in a confusing way. This also includes applications that are malware, phishing, or that carry out other harmful behaviors like unauthorized data collection.
- Social Engineering (Deceptive Content): Websites that trick users into doing something dangerous, for example, by telling them their software is out-of-date (like a fake flash update), or their device is infected with a virus.

If your website is flagged by Google Safe Browsing, you can use Google Search Console to find out why it was flagged and resolve the issue. You can then request a review of your website from Google, and if the issue has been resolved, the warning will be removed.

Phishing detection based on matching color palettes and domain parts of previously visited websites.

Chrome counts and stores the sum of the pixels of each color on all the pages you visit. Thus, if you copy the design of a login form or landing page one-to-one, it will be identified as phishing. To confuse this mechanism, it is enough to use the offset of the color palette through css.

Here is an example of javascript code that should be placed in the head section of the site, inside the "script" tag:

```
Code document.documentElement.style.cssText="filter:hue-rotate(4deg)";
```

Chrome also checks that previously visited domains and sub-domains and their parts match those that it sees in the address bar.

Thus, domains like telegram.blabla.com or att.net.fuckyou.com or ziraatbank-login.oh-my-site.com will be identified as phishing if the user has visited these sites before. The solution is to use typos in words, or not to use branded words at all, using common ones, for example, sso-login.zirat-office.com.

Chrome sharpens the mechanisms for checking for phishing if your phish contains an `<input type="password">` password input field or a file is downloaded. In order to get around this protection, you can use an alternative to the input tag and use the `contenteditable="true"` attribute on any html tag.

An example of the implementation of turning any tag into a text input can be found here:

<http://jsfiddle.net/Onojd66r/>

Mechanism for checking url-addresses and hashes of files for a match against a preloaded database of shortened hashes stored locally.

Just don't use "dirty" domains. In the context of Google Safe Browsing, a "dirty" domain refers to the domain name of a website that has been identified as potentially harmful. However, the domain name itself does not necessarily give any indication that the website is harmful.

When a website is identified as harmful by Google Safe Browsing, it's usually due to malicious content or behavior found on that site, not the actual domain name itself. That said, there can sometimes be signs in the domain name that may suggest it's suspicious, such as:

- Gibberish or Nonsense Domain Names: Cybercriminals sometimes use random, computer-generated strings of letters and numbers to register domain names. These can be a sign of a malicious site.
- Domain Names Mimicking Well-Known Brands: This is a common tactic used in phishing attempts. The domain name may include the name of a well-known brand, but with small variations in spelling or with additional words or characters. For example, "bankofarnerica.com" instead of "bankofamerica.com".
- Domains with Many Hyphens or Subdomains: Legitimate domains usually have a relatively simple structure, while malicious domains may use many hyphens or multiple subdomains to trick users. For example, "www.bank-of-america-login.com" or "www.login.bankofamerica.security.com".
- Domains Using Less Common Top-Level Domains (TLDs): While many legitimate sites use less common TLDs (like .xyz, .info, etc), a high percentage of malicious sites also use these to take advantage of their low cost and availability.

However, these are only potential indicators and not definite proof that a domain is "dirty" or harmful. A domain's reputation in Google Safe Browsing ultimately depends on the content and behavior of the site associated with that domain, not the domain name itself. And even a normally safe website could be compromised and become harmful without the domain name changing.

If a website is deemed to be malicious, Google will flag it in their Safe Browsing database, and then when users attempt to navigate to it using a browser that utilizes Google Safe Browsing (like Chrome or Firefox), they will be shown a warning page. This is to protect users from visiting websites that could harm their devices or steal their personal information.

What if domain has been already burned down.

Go to https://search.google.com/search-console/welcome?utm_source=about-page

We add our domain as a "domain resource" - the first option of two. We confirm the ownership of the domain through a record in the DNS, and on the page that appears you will see the reason for the reddening of your domain. Having eliminated it (if possible, for example, by deleting a phishing page, or a file recognized as malicious), apply for a recheck and wait. Within an hour or two, your domain will no longer be detected as malicious.

Checking hashes of executable files and hashes of their sections against a locally stored database.

Use a good crypt, obfuscate landing js files if they contain code related to cloaking and phishing. Putting the file in the archive will not help - this only makes sense if you're bypassing Chrome's total ban on downloading certain file types, such as .lnk.

Fighting with Google Safe Browsing crawler.

The google ads verification system and the GoogleSafeBrowsing crawler use the same chrome engine with a number of modifications. Including, the chromium instance is controlled via the webdriver protocol. But half a year ago, chrome fixed the last known way to detect this connection.

Filtering by ip does not make sense, because the crawler uses proxies, including up to Kazakhstan. Thus, the general advice would be to block all countries that do not interest you at the TDS level. DNS requests will still come from google subnets. However, at the moment, filtering by the names of subnets and organizations at the DNS level is not implemented in any known DNS server.

All addresses of files downloaded by users are sent to Google for their subsequent pumping by the crawler. To get around this, you can use the creation of temporary blob-urls via javascript, or binding each url to the user's ip at the server level (php), encoding the hash of the ip-address in the link, and giving the file only if it matches.

Here is an example of creating dynamic urls and attaching them to a download button:

```
Code < script >
document.querySelectorAll('.download').forEach(function(a){
  a.onclick =function( ) {
    fetch('https://url-to-download-file').then(respond = > respond.blob( ) ).then(function(blob, url, a)
{
  a = document.createElement(
'a');
  a.style.display = 'none';
  a.href = window.URL.createObjectURL(blob);
  a.download = 'desired-filename.exe';
  document.body.appendChild(a);
  a.click( );} )return false;}} )< / script >< style type=
```

```
"text / css">.download {cursor:pointer;}< / style >
```

This code will put a download handler on all buttons that have the "download" class in their styles

Chrome blocking downloads of suspicious file.

Chrome blocks downloading a file from an unfamiliar domain in four cases:

- file download not triggered by user click (sending a .click() event via javascript won't help)
- the file is downloaded from a domain that the user has not visited in the last 24 hours.
- the file is executable, or the archive containing the file contains executable files.
- the file is not signed.

This mechanism is described here:

https://source.chromium.org/chromium/chromium/src/+main:chrome/browser/resources/safe_browsing/README.md;l=143;drc=591c5c478112625a5da995afeeb6566429c04ef9

The list of files that are considered executable can be found in the chromium sources:

https://source.chromium.org/chromium/chromium/src/+main:components/safe_browsing/core/resou

[rces/download_file_types.asciipb;l=2172;drc=af17ad3f07c1d8a24381eb7669bec0c2ffb86521](https://github.com/secretsquirrel/SigThief)

Chrome does not validate the file signature, because it is demanding on network traffic. Therefore, it is important to attach to the file any signature stolen from any other white file or installer, using the SigThief utility, for example. It's free and available on GitHub:

<https://github.com/secretsquirrel/SigThief>

Chrome has a list of trusted domains that I don't know for sure. However, googleDrive and Microsoft OneDrive services definitely belong to it.

By uploading a file to OneDrive and making it "available to everyone", you can get a direct link to download the file through online services, for example

<https://metadataconsulting.blogspot.com/2019/03/OneDrive-2019-Direct-File-Download-URL-Maker.html>

This will allow you to serve the file directly, as long as it is signed. This url can also be wrapped with a url shortener like <https://t.ly/>

Also, the file name should not contain branded words that were contained in the names of popular and previously known files, for example, the words telegram, firefox and the like. Use off-topic or generic words such as Installer, homonyms, or numbers.

Domain Age.

Domain age is one of the decisive factors in the scoring system of the google safe browsing heuristic. Don't use new domains - they will burn like matches. Either use the domain shadowing technique, or use domains with a hold of at least 7 days.

Final words.

This time the thread wasn't written by me. I have only translated it from Russian to English and added some things for better understanding the problem and context. The translation may not be perfect so excuse me for that.

All credits goes to "Aels": /profile/31147-aels/

Original thread: /topic/218301/

Thank you for reading, please leave a QL or HQ post if you enjoyed it.

Stay tuned for more content.

Sincerely, LEGENDS

Sadly, he(Aels) was arrested 3-4 days ago in Germany for installing sniffer(js) . I hope he comes out soon; he's a very good-hearted guy. He has been involved in this scene since before 2009, and he taught me how to spam during my early days.

DramaKing - August 22, 2023, 5:51 am

GSB also accepts manual submissions.

https://safebrowsing.google.com/safebrowsing/report_general/ I like to make life hard for scammers.

nimda - August 24, 2023, 3:33 am

Thank you for the explanation

Friendz - September 1, 2023, 4:52 am

Very helpful and useful article.

Answers a lot of questions why my domains gets taken down sometimes almost immediately after purchase.

booboo - September 2, 2023, 1:15 pm

awesome explanation thx

johnword95 - October 25, 2023, 1:00 am

thanks for this article.

adkjgbgad - November 19, 2023, 6:52 am

Thanks! I've been wondering about this.

NickyDev - December 14, 2023, 9:36 pm

thanks man, it is so useful for me.

good luck.

Don Cardenio - December 20, 2023, 12:16 am

Quality post, thanks!

DashPHP - December 20, 2023, 12:53 am

I found this very helpful, thanks!

L3RN3R - December 22, 2023, 7:47 pm

Geez, that's a great text!

bunsys - December 28, 2023, 7:41 pm

thanks for the information

rarejarbuy - February 14, 2024, 10:09 am

Detailed explanation

SIM-5504527 - March 28, 2024, 12:30 pm

Thank you for bringing up this thread, really helpful and informative!

Stanley - April 7, 2024, 4:34 am

I'm Stanley HF's helper bot. The OP (original poster) of this thread wants everyone to see this page and has elected to bump it to the top of the showthread listing. Since HF doesn't want a ton of LQP (Low Quality Posts) this is the best method of bumping your content. There is a small ?yte fee for each bump. Any OP can bump any of their threads on HF from the linked icon on the top of their first post.

If you haven't already read the first post of this thread we encourage you to do so.

Thanks,
Stanley

coconot432 - March 5, 2025, 9:34 am

thanks for sharing bro

