ZTEdge™
By Ericom Software

Steganography Defense

# Protect Users and Networks from Malware Hidden in Images and Attached Files

## ZTEdge Web Isolation neutralizes steganographic threats that anti-virus solutions cannot detect

Steganography – stego, for short – is the art of hiding messages or code in innocent objects, files, or text. Unlike encryption, steganography does not announce itself. Since there is nothing that flags the image or object as containing additional data -- no hex-encoded strings or no bulky lookup tables – antivirus and other technologies pay them no heed. As a result, steganography is increasingly being used to conceal and deliver ransomware attack code, malicious JavaScript, downloaders, and even entire rootkits.

A classic example of a steganography-enabled attack involves malicious data concealed within an image, without changing the appearance of the image or increasing the file size. Sophisticated approaches involve manipulating individual pixels of an image to contain steganographic data. A complete file can likewise be attached to an image using, for instance, an RAR archive format, which is ignored by image display applications but can be easily extracted by a malicious actor or program.

Images are most frequently used for stenography but other kinds of files, including Excel, Word, HTML, and even network protocol files, can be manipulated to conceal stenographic data. In a variation on in-image concealment, white PNG files may be used to conceal explicit code.

## How Stenography Works

To illustrate how steganography works – and why it is so hard to detect -- let's look at one technique by which malware might be hidden in the code of an image, and why images are ideal for this purpose.

Images leverage non-executable file formats to store compressed data. To display an image, a browser uncompresses the file as it is loaded and creates the image, pixel by pixel, based on the data. Each individual pixel, however, can store more information that is actually needed to create a good representation of the image.

| Color | | | | |
|---|---|---|---|---|
| | **Binary** | **Octal** | **Decimal** | **Hexadecimal** |
| **Red** | 11111001 | 371 | 249 | F9 |
| **Green** | 01111100 | 174 | 124 | 7c |
| **Blue** | 00001100 | 14 | 12 | c |

SentinelOne

For instance, for colors that are represented using three bytes – for red, blue and green – the final four bits of each color can be used to indicate fine differences in color. These differences, however, have little or no impact on how users perceive the image. Therefore, those bits are "available" as spots where an individual can insert data or code without degrading the image – and without it being apparent to an anti-virus solution. Only a purpose-built program will "know" to look for that data and be able to read and use it.

It is worth noting that stenography is not inherently malicious. Techniques similar – or even identical-- to the one described above may be used to "mark" images or documents for tracking or other purposes.

## Steganography Attack Mechanisms

In some instances, clicking on an image containing steganographic code is sufficient to activate the code or to signal the hacker that an access channel to the endpoint is available. More often, the steganography delivers just part of the payload, which is only extracted and utilized when the image is processed.

For malicious code hidden in attached documents, PowerShell and BASH scripts are used to automatically launch the attacks when the documents are opened.

Traditional scanning technologies such as antivirus are powerless to detect steganography since the code is hidden within – and indistinguishable from – legitimate code.

## ZTEdge Web Isolation Neutralizes Detection-Evading Steganographic Threats

ZTEdge Web Isolation protects against steganographic attacks, regardless of whether they are known to be present, by scrambling hidden content and disabling embedded malicious scripts without damaging image fidelity or desired native file functionality.

When a ZTEdge user visits a website, all content including images is opened by a virtual browser located in a hardened, isolated short-lived container in the cloud, as are attachments downloaded from emails or websites. Remote browser isolation (RBI) executes all code—malicious and benign--within the container, where it cannot reach the end user machine or the information it holds.

Sampling and compression techniques are used to create safe rendering data that accurately represents the site content, including all images and functional elements, and which is transmitted to the user's browser. Users can fully interact with the site via their regular browser (and behind the scenes, a virtual browser located in the isolated container.) Hidden code within images or other site elements that may contain malicious content, however, is not accurately rendered when RBI is used in secure frame mode, and thus neutralized of any threats it may contain.

Attachments are also routed to the hardened RBI container, where they undergo content disarm and reconstruction (CDR). The files are examined for malware which, if detected, is disarmed. The document or other attachment is then reconstructed using similar techniques to those leveraged by RBI, with similar results: Extraneous code, even if it is well hidden and not identifiable as malicious, is omitted or broken, neutralizing threats from steganographic attacks.

Low-privilege user accounts and read-only file systems further secure ZTEdge RBI containers. Limited lifespans ensure that the containers are promptly destroyed along with all website content and attached files within. ZTEdge RBI containers are Linux-based, and therefore protected from Windows attacks.

Policy-based controls enable admins to vary permissions by user group, individual user, site characteristics or other parameters. For instance, file downloads may be entirely barred for sites with newly registered domain names or from social media sites.

### Defend Against Steganography Attacks with ZTEdge Web Isolation

- RBI scrambles content hidden in webpage images
- Code in attached documents is omitted or neutralized by CDR
- Steganographic code is destroyed along with short-lived, hardened RBI cloud-based containers
- Only safe rendering data for images reaches user browsers
- Attached documents are downloaded to endpoints, with desired native functionality intact

www.zerotrustedge.com

info@zerotrustedge.com

Americas:
T +1 (201)767-2210

UK & Western Europe:
T +44 (0)1905 777970

Worldwide:
T +972-2-591-1700

Follow us