How are websites defaced?

In this post, I will go over how websites are defaced, the tools used and how to do it. (Not for malicious use).

To deface websites there are a few ways to do it, first and most common way is by using an SQLi to bypass or uncover admin credentials to then login to the site as an admin. After that, they depend on a local file inclusion vulnerability which allows for files such as .php extensions to be submitted to the site (some sites only allow .jpg, .png, extensions so to bypass this you need to put shell.php.jpg, shell.jpg.php, shell.php.png and so on). After that you open the “image” (shell) in a separate tab (Right-click the uploaded image and click "copy link location", then paste the link in a separate tab) and then log into your shell, after that you select a directory such as index.html, copy and paste your .html source code into the respective field and then press “add defacement”, after this, you navigate to the home page and watch as the home page of the site has now been replaced with your .html page.

So are web defacements targeted attacks? No, at least not most of the time, most of the time the individuals are using a technique called “Google Dorking” to locate vulnerable websites that align with their dork.

Example of Google Dorks:

inurl:admin/index.php

inurl:administrator.php

inurl:administrator.asp

inurl:admin/index.php

inurl:adminlogin.aspx

As you can see the google dorks are looking for URLs that have “admin.aspx”, “Login”, “Admin panel”, etc in the URL. This tells Google to index websites with these URLs throwing back results to exposed administration panels. Most of the google dorks are outdated in terms of their design and security and a lot fall easy prey to sequel injections.

SQLi (Sequel injection) is used to bypass the login. Example of a SQLi payload is 1'or'1'='1. If one equals one, return the condition if true. So the next step is to perform a sequel injection into the login fields to bypass authentication. A SQL injection attack takes advantage of a vulnerability in a web application that allows hackers to modify the queries that are being executed on the underlying database. Web applications that directly execute user inputs as a query are those that fall prey to SQL injections. This allows attackers to execute malicious queries, also known as malicious payloads on databases.

Local file inclusion and Remote file inclusion is used to upload a shell.php file to the webserver.

Tool’s used to perform a website defacement:

Google Dork

SQLi payloads

LFI/RFI Vulnerabilities

Shell.php

File.html

Other common ways to deface websites are XSS attacks, but we won't talk about cross-site scripting in this post.

This is generally how all those Zone-H individuals deface websites. They might also DOS the websites home page while they do it. You can use slowloris for this (sudo pip3 install slowloris).

Once you know how to perform an attack you then will know how to defend from such attacks.

Zone-H: https://www.zone-h.org