Using a Malicious Proxy to Pilfer Data & Wreak Havoc

Edward J. Zaborowski
ed@thezees.net

Abstract

People surfing the web assuming that for the most part, what they are looking at is what the website served to their browser. However, that may not necessarily be the case and someone may be sitting in between you and your browser performing a man-in-the-middle (MITM) attack, changing what you get served, what gets sent to the server, and looking at everything in between. Not only can a malicious proxy allow an attacker to pilfer data such as information entered into web forms like passwords & credit card numbers, but it can also wreak havoc by assisting an attacker in doing things such as tricking users into downloading malicious content.

Malicious Web Proxy

On a daily basis many people surf the web without an idea of what information is being divulged. Businesses may have proxies set up for a variety of reason. But what if a person set up a proxy that filtered your traffic through his proxy? Now that this user has access to your surfing habits, he may be logging your passwords, profiling you and your surfing habits, and perhaps even changing what you see on the net.

A malicious proxy can be set up by virtually anyone, allowing access to your any data that you may submit through forms, user credentials for websites, cookies, and more, provided the attacker can manage to get people to use his proxy.

Becoming the Proxy

There are a few common ways one can register the proxy so as to starts intercepting traffic from the victims some of which include ARP spoofing, static configuration, and Web Proxy Auto Detection. Each method has certain advantages and disadvantages over others as discussed below.

ARP Spoofing

Utilizing ARP spoofing is one method an attacker can become the man-in-the-middle to operate a malicious proxy. While ARP spoofing can be an effective means for executing a man-in-the-middle attack, it is potentially dangerous attack in that it can potentially cause network issues. Successful ARP spoofing however, has the distinct advantage of redirecting a large amount of clients to a malicious proxy with very little effort.

Static Configuration

Static configuration would be the safest method, from a network standpoint, of becoming the man-in-the-middle for this type attack. Browsers allow for statically configuring the proxy they

use. And while this is a safe method, its major drawback is that it is not easy to implement. This method requires some kind of pre-existing access to a user's computer to be effectively utilized, rendering it somewhat impractical, particularly for large networks. Despite it's drawback this method could be hard to detect if implemented on a small scale.

Web Proxy Auto Detection

Web Proxy Auto Detection or WPAD, would fall in between ARP spoofing, and Static Configuration in terms of safety to the network, as well as ease of deployment. While the network issues that may arise from ARP spoofing may not be a problem for WPAD, one problem that does exist is a potential denial of service for any browser clients that utilize WPAD. This DoS can occur when a Proxy Auto Configuration file directs the clients to a proxy that doesn't exist. One other drawback to WPAD is that the target network needs to be configured to accept dynamic DNS updates, and the browsers themselves need to be configured to use proxy auto detection. For the latter, this is less of an issue as some browsers are configured by default to utilize proxy auto detection.

Browsers using WPAD look for a certain number of things to find a proxy configuration file. Firstly it will check DHCP (option 252) for a server hosting a Proxy Auto Configuration (PAC) file. If not found it queries DNS next recursively checking for a WPAD host on the domain. For example, if a host's domain is *workstation01321.workstations.somecompany.com*, the browser will look for *wpad.workstations.somecompany.com*, and if not found, *wpad.somecompany.com*, and so on. The browser will search until it finds the first available server hosting a PAC, and obtain the proxy information from that URL.

However, for a network configuration that fulfills the aforementioned requirements, WPAD can lead to a great success in redirecting the victims' HTTP traffic through the malicious proxy in a relatively passive manner. A secondary benefit of utilizing the WPAD method is that is provides an attacker the ability to easily specify and change a proxy host. This proxy host could potentially even lie outside of the target network.

Other Components

Once an attacker has positioned him or herself to start intercepting HTTP traffic, the hard part is over, but that was just the first step to operating a malicious proxy. Depending upon the attacker's intent, additional components may be necessary to perform this attack. These components include an HTTP server, and a proxy server. Much of these functions can be accomplished by using Apache's HTTPd server, and having a decent knowledge of Javascript, and some regular expressions.

HTTP Server

An HTTP server is an important component of this attack for a couple of reasons. The first reason is necessary to server a PAC file to hosts if the attacker is using WPAD to intercept hosts.

The second reason is to serve files to the victims, whether it be Javascript or an root-kit for the victim to download and execute.

Proxy Server

The proxy server is the most important part of the tool kit. Unlike a traditional proxy that acts as an intermediary with little or no modification to the requests or responses a malicious proxy can both filter and modify the responses & requests. Modifying responses is a selective process, modifying only those files that make the most sense to do so, for example, HTML & Javascript files. Failure to do so could cause portions of the page to render improperly, or not at all.

Once the proxy receives data that is capable of being modified, it can then use regular expressions to inject data into the HTML document, which is in turn sent to the client. The victim's browser, upon parsing and rendering the HTML activates the injected Javascript, modifying the DOM as it needs to, depending upon the desired outcome.

What about HTTPS?

Using these methods, even HTTPS can be compromised to an extent, and works much in the same way that it does for HTTP traffic. In this instance however, another server is required that acts as a the remote HTTPS server. The malicious proxy, before the request is made, replaces the information in the CONNECT request with that of a server acting as the intended HTTPS server. One key difference is that the server itself acts as a pseudoproxy, cloning the requests to the intended server, and responding in kind. It uses a self-signed certificate to match the request as to not arouse too much suspicion.

Depending upon the browser of the user doing the surfing, an alert will be generated that a browser received a self-signed certificate. The ball is now in the end-user's court as to whether or not accept the self-signed certificate. Despite the fact that this method is not as transparent as it is for its HTTP counterpart, it is another tool that can be used in an attackers arsenal.

About the Doppelganger Tool

Doppelganger was created to combine the functionality of a WPAD man-in-the-middle attack, the HTTP server, and proxy capabilities into a single utility to simplify the process of deploying the attack. Doppelganger takes a lot of the guess work out of creating and running a malicious proxy by automating, configuring, and launching the different components, with minimal effort.

Doppelganger is designed to allow an attacker to gain access to all the information contained in the web traffic and to potentially misrepresent the page that a user sees by injecting Javascript before the user's browser receives it. This is largely achieved by modifying the Document Object Model (DOM) which grants an attacker the ability to view a user's data, to manipulate the target's browser into divulging considerable amounts of information, or to add false data into an otherwise "trusted" site.

Malicious Proxy Scenarios

Scenario 1: Password Grabbing

There are two options for password grabbing when deploying Doppelganger. The simplest method simply extracts any headers containing the Authorization line. This allows an attacker to decode any Basic Authentication. In the case of form-submitted login information, using the Javascript injection, an attacker can intercept any form data before it is submitted.

Scenario 2: Root Kit Installation

An attacker has Doppelganger in place and would like to install a root kit on some systems. One way this can be accomplished using Doppelganger would to be manipulate the DOM of a web page, to hide portions or the entire page, informing a user that in order to view the page a plug-in must be installed to continue. That link would redirect to a file held on the Web Distort HTTP server which the user would then in turn download & install.

Conclusion

While users may or may not be aware of it, an attacker utilizing a malicious proxy has the ability to compromise data on a large scale, using a simple, easy to deploy attack. Whether the proxy affects a single user or a large enterprise, any data traversing from the client to the server has the chance of being divulged, risking passwords, credit card numbers, and more potentially sensitive data. Furthermore, utilizing the position the attacker has gained, he can seed the network with other files that can be used to further infiltrate the network, or exfiltrate data from the network.

Revisions:
23 March 2009 - Added HTTPS/SSL Paragraphs
24 June 2009 - Updated contact E-mail address