Fingerprinting Websites Using Traffic Analysis

Andrew Hintz
drew@guh.nu
http://guh.nu

The University of Texas at Austin

Abstract. I present a traffic analysis based vulnerability in SafeWeb, an encrypting web proxy. This vulnerability allows someone monitoring the traffic of a SafeWeb user to determine if the user is visiting certain websites. I also describe a successful implementation of the attack. Finally, I discuss methods for improving the attack and for defending against the attack.

1 Introduction

Although encryption can hide the contents of data sent on the Internet, people often forget that encryption does not hide everything. Somebody eavesdropping on an encrypted conversation can still tell who is communicating and how much data is being transferred. For example SafeWeb, an encrypting web proxy, is vulnerable to an attack that can be done by simply looking at the amount of encrypted data that is transferred. This vulnerability allows an eavesdropper, such as a government, to tell if the people it is monitoring are visiting certain websites. This vulnerability has been successfully implemented on a small network and can probably be used on an entire country. However, there are several practical methods that can be used to adequately protect against this vulnerability.

2 Definition of Traffic Analysis

The process of monitoring the nature and behavior of traffic, rather than its content, is known as traffic analysis [1]. Traffic analysis usually works equally well on encrypted traffic and on unencrypted traffic. This is because common encryption methods, such as SSL, do not try to obfuscate the amount of data being transmitted. Because of this, traffic analysis can usually tell you not only who the receiver and sender of the data is, but also how much data was transferred. In certain situations, an attacker having knowledge of the amount of data transferred can have disastrous results.

3 SafeWeb

SafeWeb is an encrypting web proxy [2]. It attempts to protect its users from both the websites they are accessing, and also from anyone who is monitoring their network connection. It attempts to hide the identities of its users from the web servers that they are accessing. It does this by not only making webpage requests on behalf of its users, but by also rewriting potentially exposing content. However, specially crafted JavaScript can be used by malicious websites to circumvent this type of protection provided by SafeWeb [3].

SafeWeb also attempts to prevent someone who is monitoring the network of a SafeWeb user from determining what the user is viewing. It uses a combination of SSL and JavaScript to encrypt webpage content and the URLs for the pages being viewed by the user. Because of this, an attacker monitoring the network connection of an end user cannot determine the actual content or URLs that the user is viewing. Although SSL adequately protects the content of the data, SSL does not adequately guard against traffic analysis.

4 Fingerprinting Websites

When a user visits a typical webpage, they download several files. A user downloads the HTML file for the webpage, images included in the page, and the referenced stylesheets. For example, if a user visited CNN's webpage at www.cnn.com, they would download over forty separate files. Each of these forty files has a specific file size which is for the most part constant.

When a user views a webpage using SafeWeb, they still download all of the files associated with the page. In a typical browser, such as Microsoft Internet Explorer 5, even when using SafeWeb, each file is returned via a separate TCP connection. Because each file is transferred in a separate TCP connection, each file is also returned on a separate port to the user's computer, and it is quite easy for an attacker to determine the size of each file being returned to the SafeWeb user. All the attacker has to do is count the number of bytes that are being sent to each port on the SafeWeb user's computer.

If someone were to monitor the network of a SafeWeb user as they visited a website, the eavesdropper would be able to determine the number and the approximate size of the files that a user received. For example, the eavesdropper would know that the user created four connections that each received respectively 10293 bytes, 384 bytes, 1029 bytes, and 9283 bytes. Each of these transfer sizes directly corresponds with the size of a certain file that was received by the user. The set of transfer sizes for a given webpage comprises that page's fingerprint.

Webpages with a large number of graphics, such as the CNN webpage, have fingerprints that are composed of many different sizes. The more files in a given fingerprint, the larger the chance that the fingerprint will be unique. Let's do a quick estimate of the number of different possible fingerprints. For mainstream sites that have a large amount of graphics, we can conservatively estimate that there are 20 different files in the page. Let's say that each of these files has a random size between 500 bytes and 5000 bytes. This means that there are approximately 4500 different sizes that each of the 20 different files can be. Raising 4500 to the 20th power gives us that there are perhaps 10 to the 73rd different possible fingerprints. This number is much, much larger than the total number of webpages currently on the world-wide-web. However, remember that the 10 to the 73rd number only applies to websites that have approximately 20 files associated with them. Websites that are purely HTML and do not reference any other files, such as graphics, would probably not have a unique fingerprint. This is because, using our previous estimate, there would be only about 4500 different fingerprints for websites that are composed of only one file. There are certainly more than 4500 text-only webpages on the world-wide-web, so not all of the fingerprints for text-only webpages are unique.

5 The Real World Threat

Several governments of the world consider the viewing of certain content on the Internet to be illegal. For example, the Chinese government considers the viewing of any dissident political ideas to be illegal [4]. Because the Chinese government controls all Internet connections into and out of the country, they can not only monitor all Internet communication with computers outside of China, but also block any outside sites. The Chinese government has blocked websites such as CNN, the BBC, and the New York Times [5]. A common use of SafeWeb is to circumvent this blocking of websites. Because SafeWeb hides both the contents and the URLs of the final site being visited, the Chinese government can not easily tell what website any SafeWeb user in China is viewing.

Using the previously mentioned file size fingerprinting system, the government could generate fingerprints for all illegal websites that it knows about. It could then watch all traffic for these fingerprints. Users whose traffic patterns sufficiently match the fingerprint for a banned website are then known to be viewing the banned websites. Although the government would have a huge amount of traffic to analyze, https traffic comprises only a very small portion of all Internet traffic. Also the government would have to periodically generate new fingerprints, because many blocked sites are news sites whose content changes frequently.

6 Implementing a Fingerprinting Attack

In order to test the feasibility of the fingerprint attack, I decided to actually implement the attack. In order to implement the attack, I first created a program that analyzes a tcpdump log and generates a fingerprint of the https traffic in the log. The program creates the fingerprint by calculating the total amount of https data that is sent to the user on each of the user's ports. The tcpdump log can be generated by any computer which can monitor the traffic being sent to the SafeWeb user's computer. The implementation of the attack and a few example tcpdump log files are available on my website at `http://guh.nu/projects/ta/safeweb/`

As an example, for a SafeWeb user visiting cnn.com on November 6, 2001, the following fingerprint was generated:

```
size:538        count:1
size:555        count:2
size:563        count:1
```

... [34 lines of data have been removed] ...

```
size:12848      count:1
size:18828      count:1
size:39159      count:1
total number of different sizes: 40
```

Size is the amount of data in bytes received on a specific port and count is the number of times this specific data size was seen in the log file.

The fingerprinting program can also determine how similar two different fingerprints are. It does this by counting the number of exact file size matches in two fingerprints. Here are the results when comparing the fingerprints of two different users visiting cnn.com a few hours apart:

```
Number of connections in the file "cnn.com": 43
Number of connections in the file "cnn.com2": 42
Number of exact matches: 32
```

Here is the output comparing a user visiting cnn.com with a user visiting bbc.co.uk:

```
Number of connections in the file "cnn.com": 43
Number of connections in the file "bbc.co.uk": 38
Number of exact matches: 2
```

Several pages were tested to verify that they could be easily fingerprinted. The pages I examined were cnn.com, bbc.co.uk, nytimes.com, slashdot.org, and washingtonpost.com. I visited each of these pages in Microsoft Internet Explorer 5 using SafeWeb and kept a separate tcpdump log of each site I visited. About an hour later, I repeated the same process with a different computer viewing the same webpages.

When I compared the fingerprints for sites that were the same, the smallest number of exact matches I found was 21. The smallest matches-to-connections ratio was 25 to 55. In other words, at least 45% of the connections were always exact matches if the two fingerprints were of the same websites. However, typically 75% of the sizes were exact matches.

These numbers only have value when compared to the number of false file size matches. I compared each fingerprint of each site to each fingerprint of each different site. The most number of false file size matches that I got when comparing fingerprints of different sites was 2. The largest percentage of false file size matches that I got on any given comparison was 6%. However for sites that were different, usually only either 0 or 1 matches occurred.

These initial results show that it is possible to maliciously fingerprint webpages on a small scale. In order to determine the difficultly of fingerprinting webpages on a very large scale, further tests need to be done. Extensive, large scale tests have not yet been done because SafeWeb has shutdown its free web proxying service [2].

7 Improving the Attack

Although the basic attack that has been described is sufficient for matching fingerprints on a small scale, the attack may need to be improved in order for it to work on a large scale. There are several things which can be done to improve the fingerprint attack against SafeWeb users.

7.1 Analyzing the Order of Transmissions

When a user visits a webpage, the first file they usually download is the HTML file for the webpage that they are visiting. The user's browser then parses through the HTML file and requests any referenced files, such as graphics and stylesheets. Each particular browser usually requests referenced files for a particular webpage in the same order. An attacker could take this into account and evaluate not only the size of the transmissions, but also the order in which they occur. In an ideal situation, looking at the order of transmissions would increase the uniqueness of a page with twenty files by about twenty factorial, or about 10 to the 26th power. However, in order to take maximum advantage of this improvement, an attacker would have to generate several fingerprints for each website, because the attacker would need one for each different web browsing program.

7.2 Improving Creation of the Fingerprint

There are several things that can be done in order to improve the accuracy of the creation of the initial fingerprint. Because each fingerprint taken inherently includes some noise, it would be beneficial to use multiple sets of data in order to generate a more accurate fingerprint. One way this could be done is to take several fingerprints of the same website as viewed from different computers. All of these fingerprints could then be added together. Any file size which occurs some minimum number of times could then be considered to be an accurate file size and included in the fingerprint which is actually used for the attack.

7.3 Expanding Fingerprints to Entire Websites

Another idea for improving the accuracy and completeness of the attack is to expand the concept of a fingerprint from just one webpage to an entire website. When a user visits a website, they often times visit several pages at the same site. An attacker could take this into account by creating a fingerprint which contains all the file sizes for all files which are available from a certain website.

For example, a fingerprint for the cnn.com website could include not only the files associated with the main page, but also all the files that are associated with any page linked to from the main page. One thing that should be noticed is that most websites share common graphics and stylesheets among several pages on the site. When a user visits multiple pages on the same website, the user often caches the graphics that they have already downloaded and therefore do not need to re-download all the graphics associated with each individual page.

7.4 Improving Matching

One way to improve matching two fingerprints together is to not require two file sizes to be exactly the same in order to have a match. For example, it could be assumed that if two file's sizes are within 5 bytes of each other then they are similar enough and are probably the same file. In order to see if this improves the attack, I implemented this range matching method. The range matching program first looks for matches that are exact. With the remaining sizes that have not yet been matched, it looks for sizes that are 1 byte apart and matches those that are. It continues this same process until it reaches the specified range. However in my test data, this range matching added roughly twice as many false positives as it did correct matches.

8 Protecting Against Fingerprinting

There are several practical methods that can be used to help protect against website fingerprinting. Some of these must be implemented by the web proxy and some can be implemented by the end user.

8.1 Adding Noise to Traffic

One way of protecting against fingerprinting is for the proxy to add extra noise to the data that it returns to the user. The methods of adding noise described here require the proxy to modify the data before returning it to the user. Although this may result in a performance hit, SafeWeb already modifies the HTML that it returns by reformatting all the URLs on the webpage that it returns to the user.

Modify Sizes of Connections The web proxy could add extra randomly sized data to the files that it returns to the user. This extra data can be made so that it does not alter the appearance of the webpages that the user views. For example, a randomly sized comment could be added into every HTML file just after the ⟨html⟩ tag at the beginning of the document. Many image formats, such as JPEG, also allow variable sized comments. The more random, extraneous data that is added to each file, the more the true size of the files will be obscured. However, adding extra data to transmissions will increase the amount of bandwidth that is required.

Add Extra Fake Connections In order to lower the percentage of connections which match a fingerprint, the web proxy could add in extra, randomly-sized connections. It could do this by inserting randomly-sized 1 pixel by 1 pixel transparent graphics into the HTML document. These extra graphics would be most effective if added throughout the file, as opposed to putting them all at the beginning or end of the file. This is because a large scale fingerprint matcher would probably look for clumps of matches, and adding extra connections to the beginning or end of a transmission would not help to break up the clump of matches. However this method has several drawbacks. First of all, adding even small graphics to some webpages could disrupt the intended layout of the page. Also, adding a significant number of sizable, extra connections would require a significant amount of extra bandwidth. For example, in order to cut the percentage of matches in half, the proxy would have to approximately double the amount of bandwidth that it uses.

8.2 Reduce Number of Files Transferred

For a quick and easy solution, SafeWeb users could choose to not view graphics on the webpages that they visit. This option is already available in most web browsers. By choosing not to view graphics, a user would drastically decrease the number of files received for most webpages. For example, if a user visited cnn.com with graphics turned off, they would download less than 25% the number of files that they would have downloaded if they had viewed the site with graphics turned on. The number of files transferred could be reduced even further by disabling things such as stylesheets and ActiveX controls. This method would make each fingerprint have a very small number of file sizes, and therefore most likely not unique. However, this would of course severely inconvenience most users because they would not be able to view any graphics on the webpages that they visit.

8.3 Transfer Everything in One Connection

Another approach to protecting against fingerprinting is to make it difficult for an attacker to determine the size of each file being transferred by lumping all the files together. There are a few methods which might make this possible. One method is for a client to not open multiple connections to the same webserver. There is a Windows registry setting for Microsoft Internet Explorer 5 which sets the maximum number of simultaneous connections to any given webserver. However this setting does not seem to have any effect when browsing the web using SafeWeb. Although this technique would make it more difficult for an attacker to find the size of each file being transferred, it may still be possible to find the size of each file by looking at the timing of the packets transferred.

Some servers have the capability to return a webpage and all associated files in a single tarball to the user. Using this method, it would be impossible for an attacker to determine the size of each individual file. However, neither all browsers nor all webservers have this capability.

9 Conclusion

Although SafeWeb is no longer open to the public[1], the ideas presented can be applied to other encrypted web proxies. The issue that the size of data is often not obfuscated by typical cryptography is something to also keep in mind in areas other than proxies. For example, there is a vulnerability in some versions of SSH where an attacker watching a connection can determine the size of the password being used. This is due to the fact that the size of the password is not obfuscated [7].

10 Acknowledgments

I would like to thank Tom Brown and Paul Sack for their very helpful comments about this paper. I would also like to thank all the people at SafeWeb for providing a very useful service to the entire Internet community.

References

1. Bruce Schneier, *Applied Cryptography* (New York: Wiley and Sons, 1996) 219.
2. http://www.safeweb.com
3. David Martin and Andrew Schulman, Deanonymizing Users of the SafeWeb Anonymizing Service, Technical Report 2002-003, Boston University Computer Science Department, February 2002. To appear in Proceedings of the 11th USENIX Security Symposium, August 2002.
4. New PRC Internet Regulation `http://www.usembassy-china.org.cn/english/sandt/netreg.htm`
5. Steve Friess, "China Re-Blocks News Sites," *Wired News* `http://www.wired.com/news/politics/0,1283,47121,00.html`
6. SafeWeb PrivaSec Alliance Press Release, August 14, 2001, `http://safeweb.com/pr_privasec.html`
7. D. Song, D. Wagner, and X. Tian, *Timing Analysis of Keystrokes and SSH Timing Attacks* (10th USENIX Security Symposium, 2001) 2-3.

[1] PrivaSec LLC has licensed SafeWeb technology, and will be providing a service called SurfSecure that appears to be a rebranded version of the SafeWeb anonymous web proxy [6].