Colonial Academic Alliance Undergraduate Research Journal

Volume 4 Article 4

2015

Reclaiming Information Privacy Online

Subrata Acharya Dr.
Towson University, sacharya@towson.edu

Sara Gorman
Towson University, sgorma2@students.towson.edu

1. Background & Introduction

The goal of this research is to identify ways in which privacy is compromised through online Internet browsing, and design approaches that mitigate the problem. Detailed evaluations on HTTP headers, tracking mechanisms, IP addresses, and encryption of data in transit would help to determine *what* and *how much* user specific information is being lost. A number of privacy preserving tools such as Ghostery, BetterPrivacy, HTTPS-Everywhere, Masking Agent, Disable HTTP Referer, and VPN will be implemented and tested. Finally, the inferences from these tests will help to demonstrate that a greater level of privacy can be obtained through the regular use of such tools.

In recent years, there has been increased concern about privacy. For example, Internet Service Providers (ISPs) collaborate with the Center for Copyright Information (CCI) to establish a system to alert users about infringement of copyrighted materials. The system is named the Copyright Alert System (CAS). The initial few alerts are warning messages, but by the fifth alert the ISPs may begin to take stronger measures to get the customer's attention. It is up to the ISPs to deliver the warnings and determine what actions need to be taken to encourage the cessation of downloading copyrighted materials. Though termination of service is not a preferable penalty by ISPs due to financial reasons, there may be other measures such as fines or throttling of bandwidth that would help in diminishing the problem. [1] Currently the participating ISPs include AT&T, Cablevision, Comcast, Time Warner, and Verizon. [2] These ISPs and the CCI are working together with a company called MarkMonitor. [3] MarkMonitor is a brand protection company that serves over half the Fortune 100 companies in order to monitor and determine which IP addresses is violating copyright laws. MarkMonitor works by monitoring major Peer-to-Peer (P2P) networks, video linking sites, cyber blogs, cyberlockers (Internet hosting services), newsgroups, auction sites, business-to-business exchanges, websites, and emails. [4] MarkMonitor is then supposed to verify that the suspected IP address is indeed violating copyright law, and then inform ISPs to notify the customer appropriately. It has been stressed that there will not be any release of personal information during the above discovery process. However, user privacy leaks have increased tremendously in the recent past due to both the lack of stringent regulatory action and the lack of any form of an accountability plan. [5]

There have been many instances where privacy has eroded for the greater good of society. A clear example is the Transportation Security Administration (TSA) system and the privacy issues that have been raised. [6]. Additionally, information is collected from stores with hidden cameras in mannequins to trace facial features in order to determine demographic information such as the age, race and gender. [7, 8] This leads to a non-transparent use, storage and

dissemination of individual private information. Moreover, there is a lack of well-defined plan of action for evaluating privacy violations and accountability in such systems. This general lack of information privacy combined with major ISPs jumping on the copyright protection bandwagon compels privacy-minded people to wonder *how much* the ISPs know about end users (or could know if they wanted to) and *how* they might manage any privacy at all in their communication system.

The aim of this research is to determine the means, by which private information could be prevented from being leaked over the Internet, gathered by trackers, sold to advertisers, or peeked at by a curious ISP or federal government. Information privacy, as with most security, cannot be absolutely guaranteed with the communication nature of the current Internet. The objective then should be to make user information harder or more time-consuming to obtain so that adversaries looking for such information will either lose interest or attack other, easier targets.

For the purposes of this research, several aspects of information privacy will be looked at: HTTP headers, cookies, trackers, Local Shared Objects (LSOs), and other ways to secure data in transit from prying onlookers. The test environment will be a Windows 7 SP 1 64-bit virtual machine (VM) with Mozilla Firefox version 16.0.2 installed as the browser; the latest versions of Java (version 7 update 9) and Adobe Flash Player (version 11.5.502.110) have been installed so that most websites will be able to have full functionality. Due to time constraints, tests on Mac and Linux environments will not be conducted, but most of the software that will be used is the evaluation are cross platform or have Mac and Linux versions available. Experiments will be conducted in an attempt to increase privacy for the above-mentioned metrics. Tools used to aid in this process include Firebug (Firefox Add-on) version 1.10, SQLite Database Browser version 2.0, Wireshark version 1.8.3, WinPcap version 4.1.2, and Collusion (Firefox Add-on) version 0.24. [Appendix A].

2. Methodology and Evaluation

The goal of this research is to test and verify the various aspects of web browsing, detail how it compromises a user's online privacy, and, finally demonstrate steps to implement the tools that would aid in the prevention of data loss and improve privacy of end-users. The aspects of web browsing that commonly lead to the unveiling of user's data includes: HTTP headers, cookies, trackers, Document Object Models (DOMs), LSOs, and unencrypted traffic. HTTP headers are in use every time a user browses the web. For example, when a web search is conducted, the search browser makes an HTTP request to a remote web server. This request

can take the form of a GET, POST, or HEAD request. GET is the most common request type. It is usually used to retrieve html, images, JavaScript, CSS, etc. POST requests are often used when information needs to be sent to the web server, like when an online form is being filled out. HEAD requests are similar to GET requests except that they only return a status code and a short message so that the browser can tell if the requested page has been modified, cached, or has an error like 404: Page Not Found. [9]

Figure 1: Get Request Example

Figure 1 represents the header contents of a GET request for a Google search on the word “Firefox”. All that the header needs in order to display the web page request is the method (GET, POST, or HEAD), path (address of the requested content), and protocol (usually HTTP). Everything else in the header is additional information about the user’s browser, what site the user last visited, preferred language, types of encoding the browser will accept, and cookie information. As can be seen in Figure 1, the “User-Agent” value contains a good deal of information about the browser and operating system of the user. The requesting machine is running Windows 7 based on the Windows NT 6.1 platform; the WOW64 value indicates that there is a 32-bit browser running on a 64-bit operating system; and the browser is version 16 of Firefox (MSDN). The Accept-Language value indicates that the user’s preferred language is English. The Referer value tells the server where the user was on the web when it made the request. The Accept value defines what schemes will be accepted as a response to the request. Finally the DNT value, which is not by default enabled, is for the do-not-track option available in Firefox (W3C).

HTTP headers contain a lot of information that can be pieced together to get a general picture of the user and what software they are running. While none of this information is particularly critical, it does contribute to an overall picture of the end-user and reduces privacy. There are a number of Firefox add-ons that can be used to limit the information the header gives away to the web server. Besides using add-ons, another way to protect this information is to use HTTPS

whenever possible, as it will encrypt the data in transit so that third parties may not examine the traffic while it is being transmitted to the destination.

HTTPS stands for HTTP Secure and uses the Secure Socket Layer (SSL) to send encrypted data between a user and a web server. SSL uses public key encryption to secure the data being sent. Public key encryption defines that the server has both a public key and a private key. The server sends the user the public key, which helps the user encrypt the data and send it to the server; the data can only be decrypted by the server's private key. Apart from encrypting the transmitted data, SSL also includes a fixed-length message digest. The server can compare the message digest to the actual message; if they match up then the server can be reasonably sure that the message has not been altered by a third party before being received. Finally, SSL can also use certificates, although in practice they are not always implemented due to cost restrictions. A certificate also serves as a digital document that is verified by a trusted third party, such as VeriSign or GeoTrust, and certifies that the server is valid. [10] This adds an extra layer of trust between the user and the server.

SSL creates an encrypted session between the user and the client that enables secure transmission of the data along with the HTTP header. Using the HTTPS version of a website whenever possible would be a good step in limiting how much of the user's information is transmitted in plain text over the Internet. By combining the use of HTTPS with Firefox add-ons it is possible to regain a certain level of privacy. Two sets of evaluations will be conducted to test these ideas. First a *preliminary* test will be run on the VM using Firebug to view the HTTP headers and next a *privacy* test will be conducted. The privacy test will be conducted with additional functionalities installed as Firefox add-ons. The results will be compared to demonstrate the effectiveness of the tools and approaches.

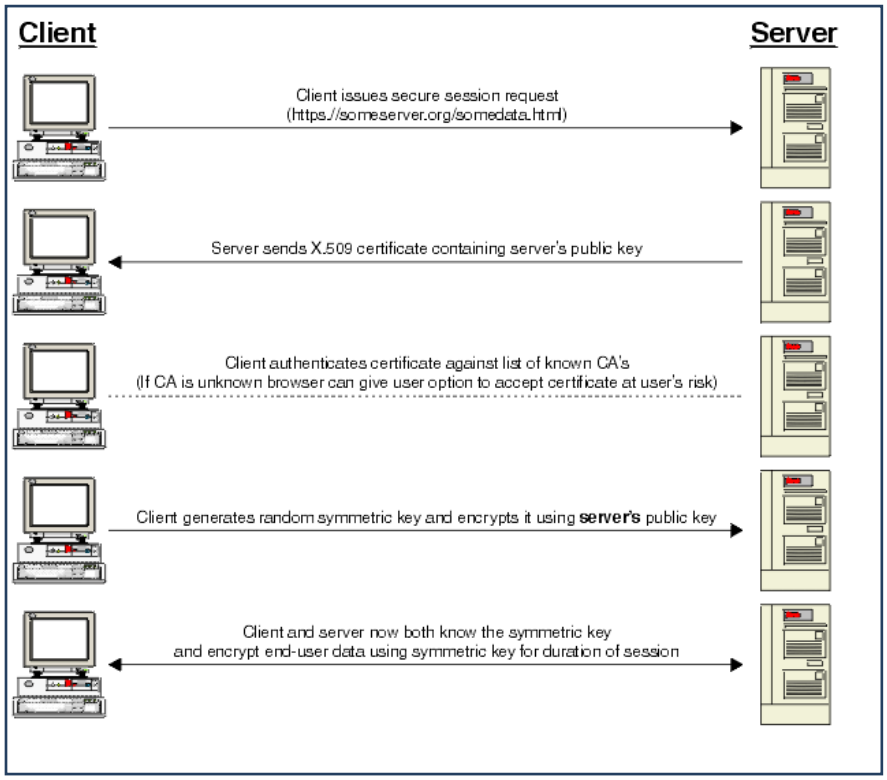


Figure 2: SSL Diagram (SSL Handshaking with Server Authentication)

2.1 HTTP Request Header: Preliminary Test

In this test, first the header contents of three HTTP GET requests will be examined through the use of the Firefox add-on and Firebug. Firebug is a handy tool that runs in the browser and performs a variety of functions, such as displaying all the HTTP requests made by a given web page and the respective header content. The purpose of this initial test is to visit a few common websites and view a typical GET request header in order to attain a general idea of what kind of information is being passed to web servers in any normal operation. The first site visited was the home page for Amazon.com. The header appeared as follows:

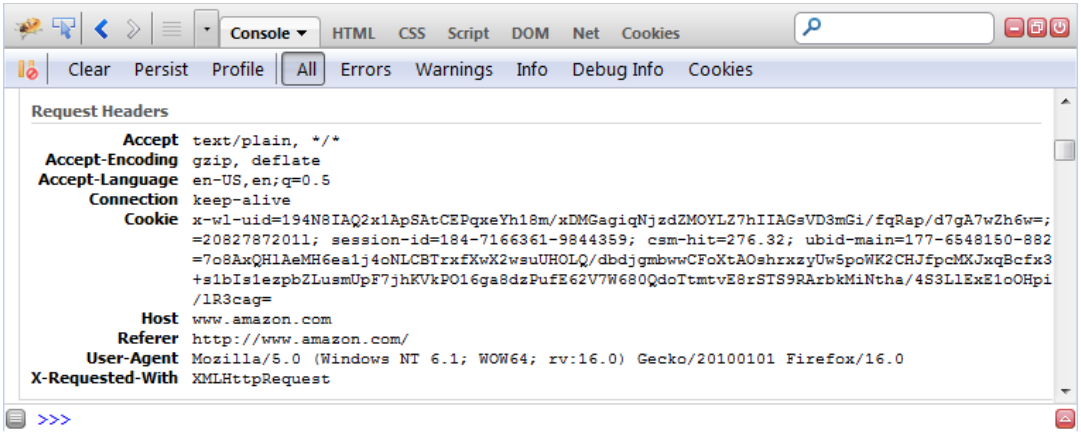


Figure 3: GET request header from Amazon.com

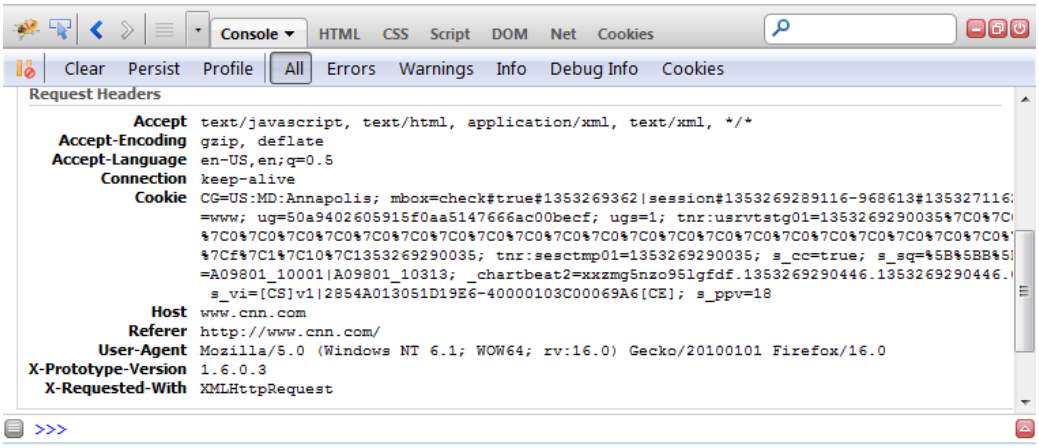


Figure 4: GET request header from CNN.com

Finally there is a request header for a Google search on the word *Firefox* as depicted in Figure 5.

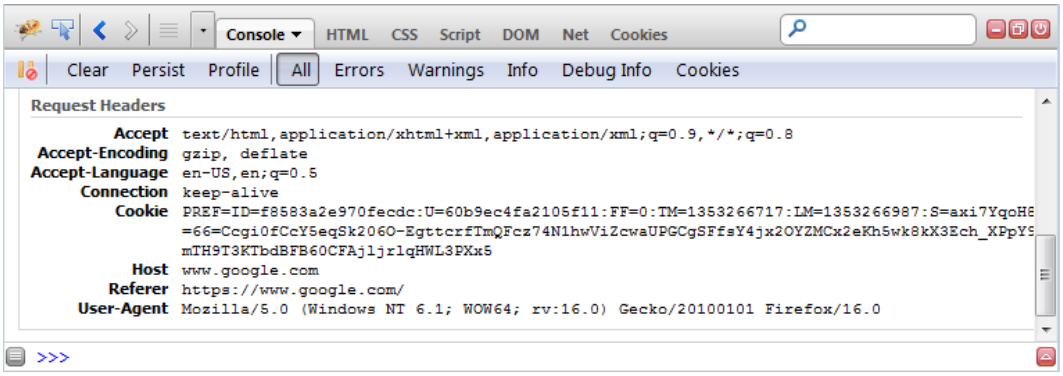


Figure 5: GET request header on Google.com (search for “Firefox”)

The HTTP request headers have a few things in common. First, they send information on where the request originated in the Referer value. They also use cookies and display browser and operating system information through the User-Agent value. Cookies contribute considerably to user privacy loss via tracking. Privacy is increased by: a) avoiding being tracked through the Referer value; b) limiting the data in User-Agent, and 3) using HTTPS more often, in order to prevent third parties from peeking at the header information as it is in transit to the designated web server.

2.2 HTTP Request Header: Privacy Test

From the preliminary test it is clear that while most of the information in the request header is not absolutely necessary it is still included in the requests. To increase privacy there are several Firefox add-ons that will be installed. Some of them are: Disable HTTP Referer at Startup version 0.0.2.rev5 to limit servers knowing the origin of the request; Masking Agent version 1.2.0 to hide OS and browser information and HTTPS-everywhere to encourage the use of SSL where supported. Disable HTTP Referer should completely remove the referer name and value from the headers. To ensure that Disable HTTP Referer measures up to the standards of this test, the tool was enabled and the same three web sites as in the preliminary test were visited. Not a single header had the referer name or value in it.

Prior to conducting the tests for the Masking Agent tool, the Disable HTTP Referer tool was disabled through the Firefox Add-on menu to prevent any interference with this next set of experiments. The Masking Agent tool attempts to hide user OS and CPU information from the User-Agent field. The tool is installed and evaluated on the same three sample websites as above. The User-Agent still appears in the header; however upon closer inspection both the

Windows NT 6.1 and WOW64 information is replaced by masking-agent. It may not hide all of the User-Agent information but this is a fair compromise since many websites need to know what browser is running so that they may render the content in a way that the specific browser can handle.

Also tested for this part of the experiment is HTTPS-everywhere. Prior to evaluating, the previously installed Firefox Add-ons have been disabled. HTTPS-everywhere is a tool that will encourage the user's browser to use HTTPS if it is available. Some websites will offer both HTTP and HTTPS versions, but may default to the non-secure variation. This add-on will help ensure that HTTPS will be taken advantage of if it is an option. The final test results on the HTTP header request with all three of the tools enabled together and can be seen through the screenshots below.

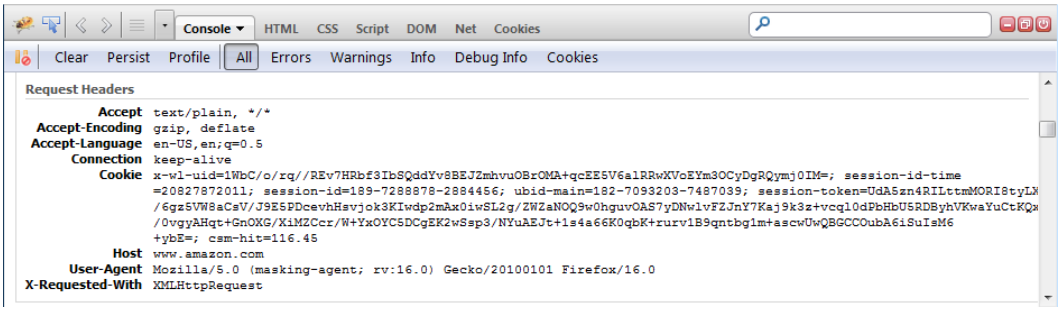


Figure 6: HTTP header request from Amazon.com

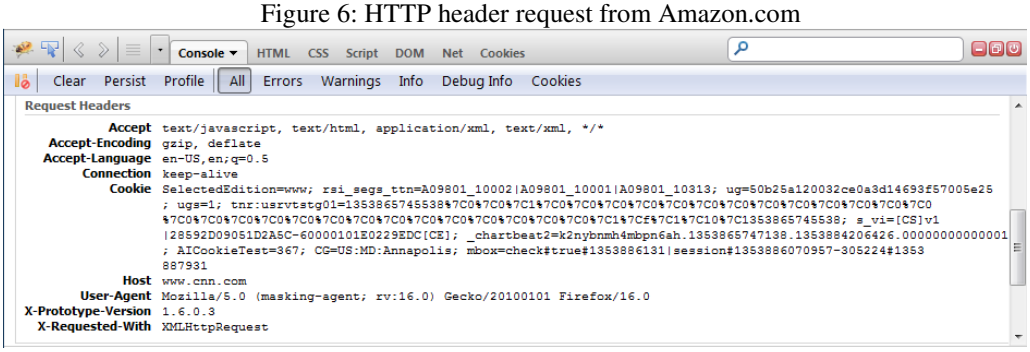


Figure 7: HTTP header request from CNN.com

Figure 8: HTTP header request from Google.com (search for "Firefox")

Finally, Google.com with a search for Firefox is depicted in Figure 8. With the Disable HTTP Referer enabled none of the headers contained the Referer name. Also the words masking-agent has replaced the OS type field. It should also be noted that using HTTPS-everywhere Google searches are redirected to encrypted.google.com.

2.3 Tracking Mechanisms

Persistent tracking methods used by advertisers and websites account for a significant amount of privacy loss for the average user. Over the years advertising companies have discovered that the Internet is not only an advertising platform but also contains a vast amount of information about users and their online behavior. According to an article in the *New York Times* there are five areas in which advertising companies collect data on users: pages displayed, search queries entered, videos played, advertising displayed, and finally advertisements served on pages anywhere on the Web by advertising networks owned by the media companies. [11] These companies track user activity not only on the website where the initial advertisement is located, but also across multiple websites through their networks. The information gathered about users includes not only their IP address, but also the website the user visited and also any other types of advertisements being displayed to the user during the time of the visit.

One way that advertising companies track online activity is via the use of cookies. Cookies are pieces of information in text format that are downloaded to the user's computer. Sometimes cookies are used by the websites to preserve user preferences so the user does not have to re-enter them on the next visit. Advertising companies, on the other hand, store cookies on the user's computer to track them across the web. For example the company DoubleClick is able to store a cookie on a user's machine at one site and then open that cookie again at another site, thus tracking and storing information about the user across multiple sites. [12] Another tracking device used frequently is called the Web bug. It

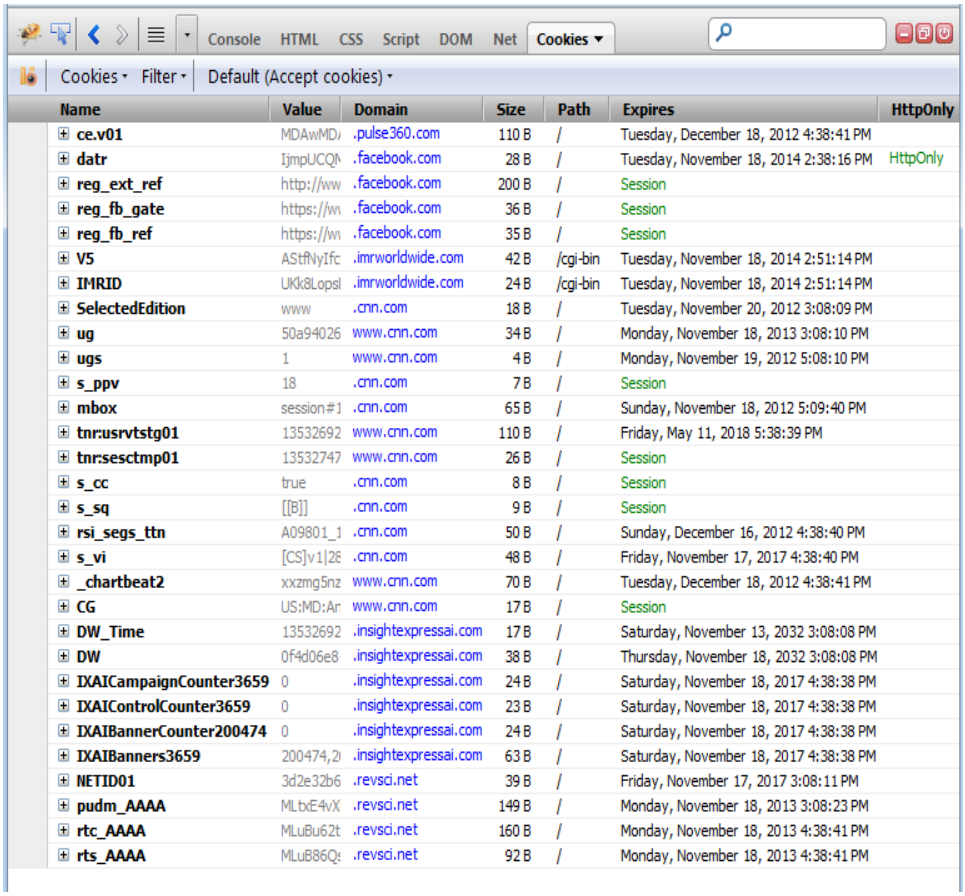
works similar to the concept of cookies and enables user information to be passed to a third party. Web bugs are very small, only one pixel by one pixel essentially making them invisible to the end user. They keep track of who is viewing a web page or email that contains the bug by sending information to a server. The information that a bug sends can include the user's IP address, the URL of the webpage where the bug is, the URL of the image that the bug is in, the time of viewing, the browser type of the user, and whether or not there is a previously set cookie for it. Given all those little bugs can do it is no surprise that advertising companies like to use them to add information to a personal profile of what sites a person is visiting. [13]

Aside from cookies stored in text files and hidden web bugs, most popular Internet browsers support what is called document object model storage (DOM). This will show up in Windows 7 under the C:\User\<user>\AppData\Roaming\Mozilla\Firefox\Profiles directory. The files will appear as .sqlite and can be viewed with an SQLite database browser. Websites are able to store data in these formatted DOMs as name/value pairs. DOMs allow for the storage of megabytes worth of information in an easy to access format but are still vulnerable to all the same tracking issues as above. [14] Furthermore, there exists another type of persistent storage, namely Local Shared Objects (LSOs). These are used by the Flash Player and contain data from them. LSOs can store fairly large amounts of information and are cause for privacy concerns because, unlike cookies and DOMs, they do not have an expiration date. Needless to say, advertisers have found ways to utilize this storage type to profile a user's activities. LSOs have a file extension of .sol and are not located with all of the other cookies but as their very own Flash Player directory under C:\User\<user>\AppData\Roaming\Macromedia\FlashPlayer\#SharedObjects\. Since, they are managed by the Flash application and not directly by the web browser, there is no simple way to manage them from the Firefox options panel.

To start the evaluation, a pre-determined series of websites will be visited to accumulate cookies, DOMs and LSOs. Web bugs will not be part of the preliminary test since they cannot usually be seen. Firebug makes viewing cookies and DOMs easy, but LSOs will have to be viewed by navigating to the directory where they are stored. In order to evaluate the tracking extent of a user's Internet browsing, a Firefox add-on called Collusion will be used to generate a graph that represents how movement across the Internet is being tracked. For the privacy test, the Firefox add-ons Ghostery and BetterPrivacy will be installed and the methods used for the preliminary test will be re-run to demonstrate how the tools enhance privacy.

2.4 Tracking Mechanisms: Preliminary Test

The test VM is enabled on the Firefox browser and ensures that all add-ons besides Firebug have been disabled prior to web browsing. The websites that will be browsed for this test are as follows: Amazon.com, CNN.com, Facebook.com, Towson.edu, Pinterest.com, Reddit.com, Google.com, and YouTube.com. The choice of sites visited and search/view options of sites are to simulate a reasonably well-rounded browsing session. In Figure 9, note that there are some cookies from websites that were not on the list of those visited. They are most likely from advertisements. All of the cookies are either session cookies, which are erased when the browser is closed, or have been given expiration dates.

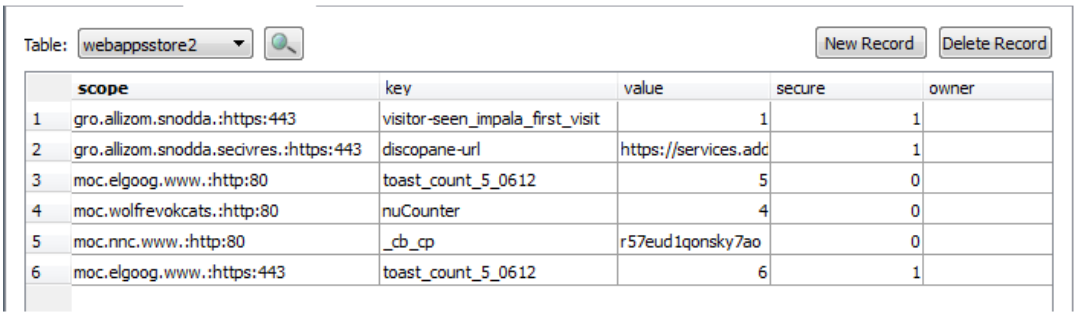


Name	Value	Domain	Size	Path	Expires	HttpOnly
ce.v01	MDAwMD/	.pulse360.com	110 B	/	Tuesday, December 18, 2012 4:38:41 PM	
datr	IjmpUCQN	.facebook.com	28 B	/	Tuesday, November 18, 2014 2:38:16 PM	HttpOnly
reg_ext_ref	http://ww	.facebook.com	200 B	/	Session	
reg_fb_gate	https://w	.facebook.com	36 B	/	Session	
reg_fb_ref	https://w	.facebook.com	35 B	/	Session	
V5	AStfNy1fc	.imrworldwide.com	42 B	/cgi-bin	Tuesday, November 18, 2014 2:51:14 PM	
IMRID	UKk8Lopsl	.imrworldwide.com	24 B	/cgi-bin	Tuesday, November 18, 2014 2:51:14 PM	
SelectedEdition	www	.cnn.com	18 B	/	Tuesday, November 20, 2012 3:08:09 PM	
ug	50a94026	www.cnn.com	34 B	/	Monday, November 18, 2013 3:08:10 PM	
ugs	1	www.cnn.com	4 B	/	Monday, November 19, 2012 5:08:10 PM	
s_ppv	18	.cnn.com	7 B	/	Session	
mbox	session#1	.cnn.com	65 B	/	Sunday, November 18, 2012 5:09:40 PM	
tnr:usrvtstg01	13532692	www.cnn.com	110 B	/	Friday, May 11, 2018 5:38:39 PM	
tnr:sesctmp01	13532747	www.cnn.com	26 B	/	Session	
s_cc	true	.cnn.com	8 B	/	Session	
s_sq	[[B]]	.cnn.com	9 B	/	Session	
rsi_segs_ttn	A09801_1	.cnn.com	50 B	/	Sunday, December 16, 2012 4:38:40 PM	
s_vi	[CS]v1|28	.cnn.com	48 B	/	Friday, November 17, 2017 4:38:40 PM	
_chartbeat2	xxzmg5nz	www.cnn.com	70 B	/	Tuesday, December 18, 2012 4:38:41 PM	
CG	US:MD:Ar	www.cnn.com	17 B	/	Session	
DW_Time	13532692	.insightexpressai.com	17 B	/	Saturday, November 13, 2032 3:08:08 PM	
DW	0f4d06e8	.insightexpressai.com	38 B	/	Thursday, November 18, 2032 3:08:08 PM	
IXAICampaignCounter3659	0	.insightexpressai.com	24 B	/	Saturday, November 18, 2017 4:38:38 PM	
IXAIControlCounter3659	0	.insightexpressai.com	23 B	/	Saturday, November 18, 2017 4:38:38 PM	
IXAIBannerCounter200474	0	.insightexpressai.com	24 B	/	Saturday, November 18, 2017 4:38:38 PM	
IXAIBanners3659	200474,2	.insightexpressai.com	63 B	/	Saturday, November 18, 2017 4:38:38 PM	
NETID01	3d2e32b6	.revsci.net	39 B	/	Friday, November 17, 2017 3:08:11 PM	
pudm_AAAA	MLtxE4vX	.revsci.net	149 B	/	Monday, November 18, 2013 3:08:23 PM	
rtc_AAAA	MLuBu62t	.revsci.net	160 B	/	Monday, November 18, 2013 4:38:41 PM	
rts_AAAA	MLuB86Q:	.revsci.net	92 B	/	Monday, November 18, 2013 4:38:41 PM	

Figure 9: Cookies from sites visited

There was only one LSO saved, videostats.sol, and it is most likely from the video that was played on YouTube. As with DOMs a normal user's browsing session would have more LSOs to view.

Figure 10 is a screenshot from the SQLite browser. It is to be noted that not many DOMs were accumulated during the test browsing session. Browsers that see more frequent use will have a considerable number of DOMs.



	scope	key	value	secure	owner
1	gro.allizom.snodda.:https:443	visitor-seen_impala_first_visit	1	1	
2	gro.allizom.snodda.secivres.:https:443	discopane-url	https://services.adc	1	
3	moc.elgoog.www.:http:80	toast_count_5_0612	5	0	
4	moc.wolfrevokcats.:http:80	nuCounter	4	0	
5	moc.nnc.www.:http:80	_cb_cp	r57eud1qonsky7ao	0	
6	moc.elgoog.www.:https:443	toast_count_5_0612	6	1	

Figure 10: SQL Browser screenshot from sites visited

Next, the Collusion add-on is run while browsing the same website list as previously mentioned. This add-on simply monitors what sites a user visits while it is running and then it makes connections between what trackers are being sent user data and how those trackers are connected to other sites that are visited. The glowing blue circles in Figure 11 indicate sites that were directly visited during the test, while yellow circles indicate sites that were not explicitly navigated yet information was sent to them. The gray connecting lines show where data was sent back and forth. These results clearly indicate that the user's privacy has been violated.

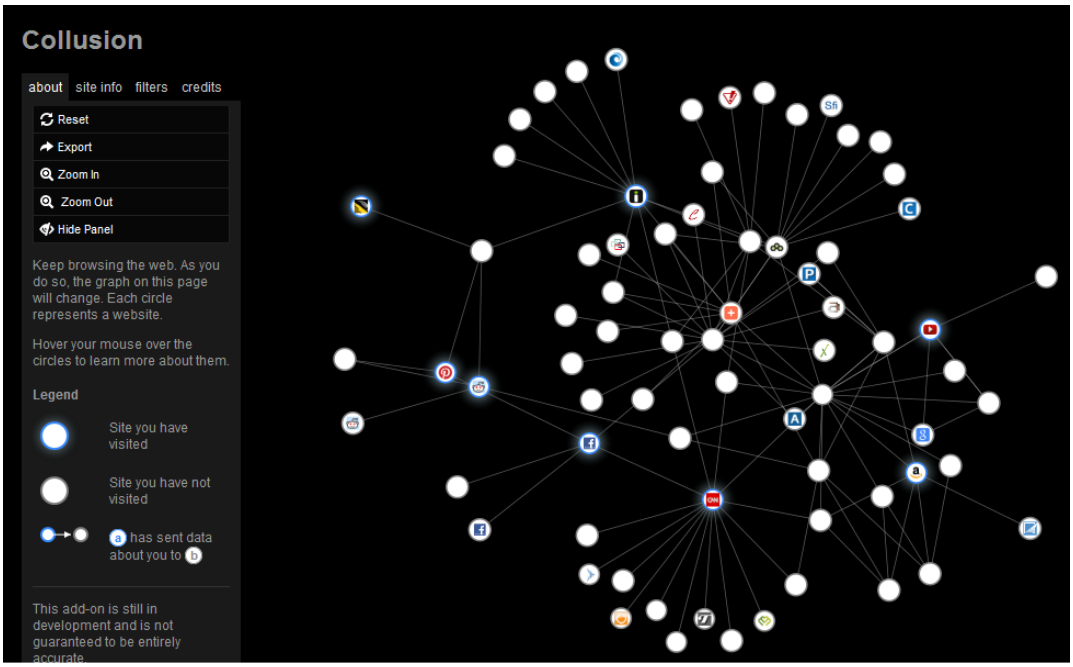


Figure 11: Collusion graph from sites visited

2.5 Tracking Mechanisms: Privacy Test

The plan for the privacy tests is to run additional Firefox add-ons that will help give users their privacy. The first tool that will be tested is Ghostery. [15, 16] This tool alerts the user about the presence of web bugs and cookies. An added feature of Ghostery is that it also gives the option to display more information about the trackers. It should be mentioned here that the Ghostery software is created and owned by Evidon, which is a company that enables businesses to access the cookies and other tracking activity on their websites. [17] It is more than likely that Ghostery's optional GhostRank data is being used not only to help improve Ghostery but to also help the advertising companies. [16] Although this is not a direct privacy concern, users may want to opt out of GhostRank. Our evaluation concluded that Ghostery is an excellent tool for uncovering, blocking, and educating users about the cookies and web bugs that may be tracking them on any given site. Ghostery's evaluation of CNN.com uncovered fourteen web bugs. The Edit Blocking Options button helps users to check the items they would like to block.

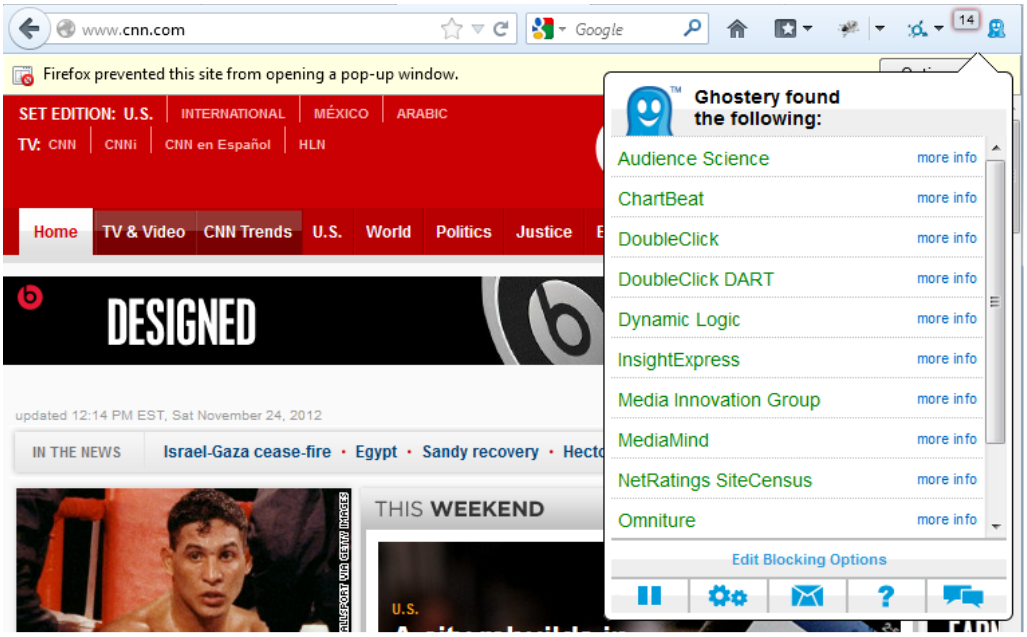


Figure 12: Ghostery results from visit to CNN.com

The next tool installed is BetterPrivacy. This is another Firefox add-on, which deletes any LSOs that have accumulated during a browsing session. When the browsing is terminated, the user is presented with a pop-up box asking if he would like BetterPrivacy to delete LSOs that were accumulated. To complete the privacy test, both BetterPrivacy and Ghostery were enabled and the same web sites as in the preliminary test were navigated. Ghostery received a thorough workout as it found many bugs and cookies, and BetterPrivacy deleted one LSO upon closing the browser window. The locations where cookies and DOMs were stored were cleared before the test through the use of CCleaner and the LSOs were deleted manually.

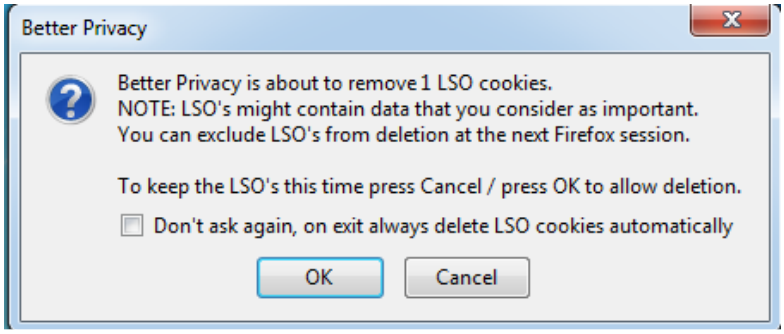


Figure 13: BetterPrivacy screenshot for removing LSO cookies

After the test CCleaner found no additional cookies or LSOs in the Flash Player directory. As in the preliminary, test the Collusion add-on was running in the background while browsing the selected web sites. The graph looked different in this test run as depicted in Figure 14. Despite having Ghostery enabled, a few trackers were sent information. However, there were considerably fewer than before, and the graph is split up into five separate groups of connections. There was a large reduction in the amount of information being shared and tracked across multiple web sites. Ghostery and BetterPrivacy may not be a perfect solution but Collusion clearly shows that there is merit in running them to attain a higher privacy level.

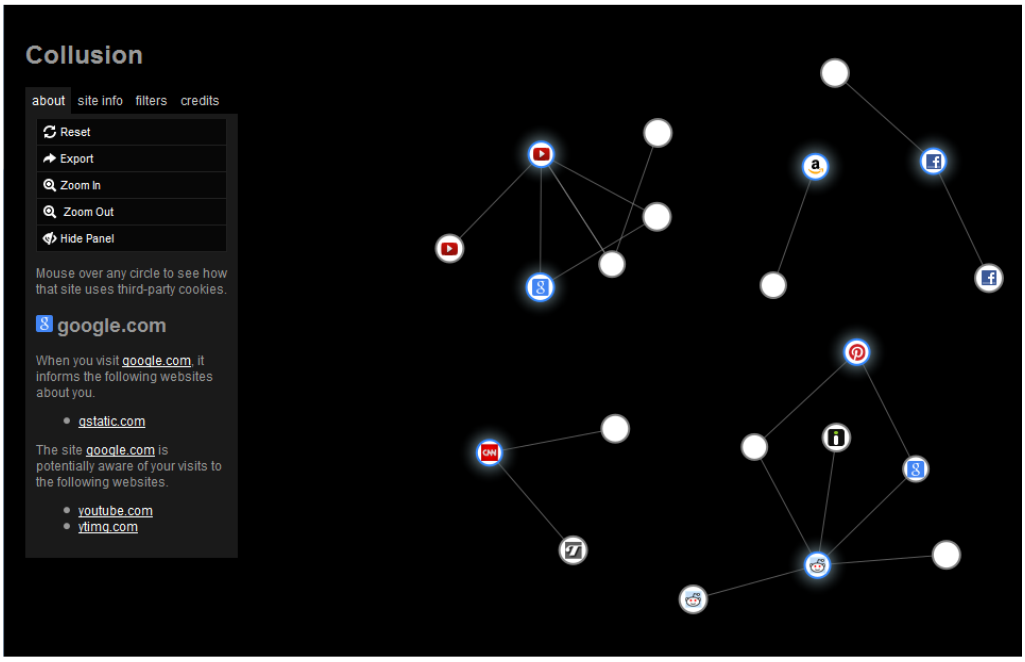


Figure 14: Collusion graph after tools installed

2.6 IP Address & Encryption

In order to ensure privacy from ISPs, or other interested parties, data needs to be encrypted as it travels from the user's computer to the destination. The user will also need to find a way to obfuscate their actual IP address so that it is harder to identify who and where they are in the Internet. One of the most common types of data that is tracked is a user's IP address. An IP address can be extremely useful in identifying a user's geographical location. Hiding an IP address may be one way to increase anonymity, but it does nothing for hiding the contents of the user's traffic while in transit. Therefore encrypting data in motion is a crucial part

of privacy. If Internet traffic is not encrypted, any snooping third party can easily see the entire contents of the packets being sent. This is especially significant for those who use Internet cafés or hot spots that provide public access to Wi-Fi. Anyone who is on the network with the right tools can sniff traffic belonging to others on the network and if the traffic is not encrypted then it can be easily viewed.

Solving the problem of encryption and IP address anonymity is best-achieved one of two ways: Tor or VPN. Tor is able to encrypt data and send it through the Tor network of random computers so that the end destination cannot see or even easily trace the traffic back to the original user. The Tor network functions by creating a series of encrypted connections through relays on the network. [18] Each relay is only aware of whom it is receiving from and where it is sending. Different users of the Tor network are likely to use different paths or circuits. The originating client constructs the circuit using the router's public key. During construction a shared symmetric key is agreed upon with each router. The Tor (The Onion Router) system got the name onion routing because each packet sent by the user is encrypted once for each router in the path and each router along the path peals back one layer of encryption and sends it on. Packets sent by a server to the user are encrypted once at each router and then the user is able to decrypt all of the layers to access the data inside. The various layers of encryption not only help to conceal the information in the packets but also the header information about where each packet is coming from and going to. [19] The method of layering encryptions and bouncing packets all around the Tor network provides the security and anonymity it touts, but greatly contributes to the slowness from which this type of network inevitably suffers. The lack of speed is one of the major drawbacks to Tor. Another drawback is that anyone can volunteer to be a Tor router. This means that there needs to be a certain level of trust between the user and all the other users who are routing the traffic. This causes some people to worry that an unscrupulous exit node will be examining all of their data since by the time the packet leaves the exit node it is no longer encrypted.

Besides using a setup like Tor, users can use what is called a Virtual Private Network (VPN) to enhance their privacy online. VPNs are able to connect remote servers and users together through a tunneled network that emulates a point-to-point link. [20] Tunneling is how information is transported in VPN systems. It works by encapsulating data being sent in a new type of package called an IP Packet, which is then sent over the network to the VPN server where they are unpacked from the IP Packet back into their original form. The contents of the message are then sent to their intended destination and the sender's address on the message will be that of the server instead of the user's IP address. [21] There are a number of different tunneling protocols that VPN services use. Two

of the most popular are PPTP VPN and OpenVPN. PPTP VPN is an extension of the point-to-point protocol (PPP) and is a commonly used for VPN tunneling. It functions on the data link layer of the OSI model while OpenVPN is on the data link and network layers. [21] OpenVPN is one of the more secure protocols as it uses SSL and hence is generally the choice for VPN users. Whether to use Tor or VPN for privacy will likely depend on user needs and affordability. VPNs generally tend to have a price tag attached (there are free versions, but they are restrictive), but they can also offer faster speeds and perhaps even a little more security. Tor on the other hand is free and can be quite slow simply because of its nature. For the purposes of this paper and its experiments the tool of choice will be VPN.

2.7 IP Address & Encryption: Preliminary Test

The preliminary test is via a packet capture with Wireshark and a visit to Cloakfish.com to see what information is displayed during Proxy-analysis. The purpose of sniffing traffic with Wireshark is to view the overall state of the traffic. The packet capture is conducted during a brief browsing session to the same site list used for the tracking mechanisms tests. The browser add-ons that were researched in the previous two experiments will be enabled during this test since the point of this particular test is to continue to improve on the privacy that has been achieved until this point. After all the sites on the list have been visited, Wireshark will be shut off. The final step is to use the Firefox browser to go to Cloakfish.com and use the Proxy-analysis tool.

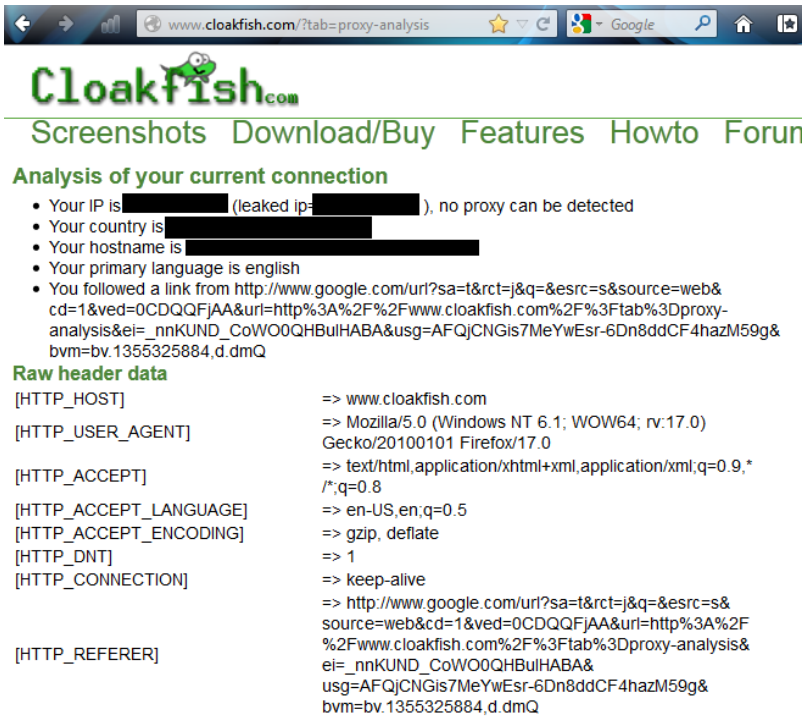


Figure 15: Cloakfish.com analysis screenshot

The Cloakfish.com tool examines the data provided to it by the contents of a packet. The packet capture revealed that most traffic is not encrypted. The source and destination IP addresses clearly indicate that the test VM is sending data and the IP addresses listed, as destinations are those of the websites that are being visited. Although Wireshark only ran for a brief amount of time it captured a distinct picture of the overall web browsing. Some information was blocked by the add-ons from the previous experiments (i.e. the User-Agent value did not reveal the OS type), but it identified the IP address and geographical location.

2.8 IP Address & Encryption: Privacy Test

Since it was determined previously that VPN provides a slightly better level of privacy and speed than Tor, VPN is what will be used in our attempt to improve privacy. The primary reasons to pick a particular VPN service is the provider's privacy and logging policy. The VPN service chosen for this experiment is TorGuard. They keep a reasonable amount of logs, delete them after 24 hours, and have a straightforward privacy policy.

Added benefits of TorGuard are the support for legal torrents and exit servers in multiple countries to increase content availability for users. Since some

of those servers are based in the US it is possible to watch Hulu and listen to Pandora over VPN. If the user does not want to purchase this service with a credit card that gives out personal information, TorGuard does accept payments in Bitcoins that are an anonymous online currency much like cash in the physical world. To conduct the test TorGuard was installed on the test VM and enabled. The exit server that was chosen was in Canada. Wireshark was set to begin capturing packets, all the add-ons were enabled, and the list of web sites that were used in the preliminary test were visited again. After the browsing session was completed Wireshark was turned off and the browser was navigated to Cloakfish for the proxy-analysis.

The results of the packet capture showed a significant difference from the preliminary test. Instead of the packets being mostly unencrypted, they were now all encrypted. Since the test machine was sending the packets, the sender information still contained the test VMs IP address, but all of the destination addresses had been changed to that of the VPN server. Since the traffic is encrypted the ISP cannot tell what the content is or determine where it is ultimately going since the encrypted traffic is going to a VPN server. To ensure that the VPN service is indeed hiding the original IP address of the user, the results of the Cloakfish proxy-analysis were examined. The IP address no longer indicated the true IP of the test machine or the geographical location as depicted in Figure 16.

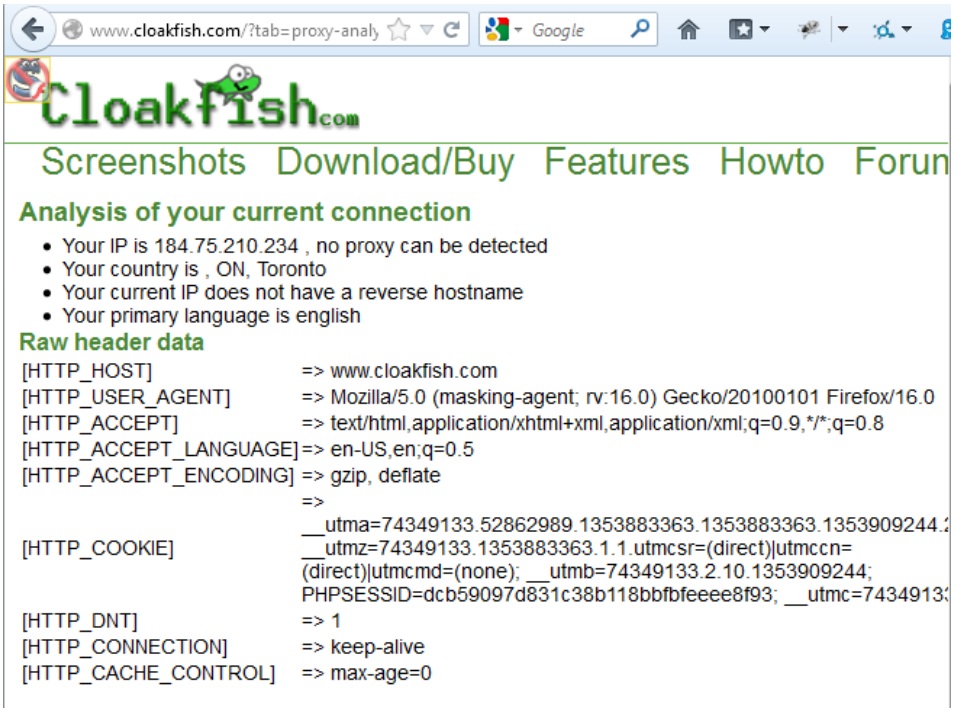


Figure 16: Cloakfish.com Privacy Test screenshot

3. Inference

The above evaluations prove that it is possible to reclaim a significant level of online information privacy. The first privacy issue covered is that of HTTP Headers. The results showed that less information is being distributed with the use of Disable HTTP Referer, Masking Agent, and HTTPS-Everywhere. In addition to extra privacy, the HTTPS-Everywhere tool can also help prevent SSL stripping attacks. In this type of attack the hacker will try to strip out the "s" from the HTTP request. Due to the fact that HTTPS-everywhere adds that in and forces sites to use the secure version whenever possible, it can help to reduce the risk of this type of attack.

The second experiment looked at the privacy issues surrounding tracking mechanisms. The result did not find an alarming amount of cookies, DOMs, or LSOs that showed up during the preliminary test, but as with most things it is what could not be seen that proved to be the biggest problem. The Collusion add-on showed how trackers on web sites were tracking users. Everything shown on the preliminary graph was connected to everything else, which indicated that the user's movement is tracked everywhere they visit and followed to other sites as well. The Ghostery add-on also revealed a whole host of trackers that were not

immediately apparent by simply looking at the web pages or viewing the data that is stored on the test VM. Installing and enabling Ghostery and BetterPrivacy demonstrated that the user tracking is reduced and there are fewer DOMs and no LSOs stored on the machine.

The third and final privacy issue addressed is that of encrypting traffic and hiding the user's original IP address. This test used Wireshark to determine if the traffic was encrypted, who was sending it, and where it was heading. Cloakfish's proxy-analysis tool helps to get an idea of what information web sites are about to obtain about the user and their IP address. After installing and enabling TorGuard's VPN service, the test results showed that the traffic was securely encrypted so that sniffers could not view the content and that the destination only revealed the VPN server and not the address of the traffic. The proxy-analysis tool was no longer able to identify the test VMs IP address or physical location of the VPN in use. The results of all the evaluations clearly indicate that there is significant improvement in privacy by implementing the tools.

4. Conclusion

This research provided a method to use strong foundation tools that can improve privacy of online browsing. The tracking mechanisms of the various tools can be used to protect against web bugs, trackers, and excess cookie files. Ghostery and BetterPrivacy are important tools and were proven to increase privacy. Ad-Block Plus and No Script Firefox add-ons could be used with the above tools. Additionally Firefox has a do-not-track option that can be enabled to indicate that a user would like to opt-out of being tracked. Finally the Collusion plug-in was executed with the previously tested tools and those recommended above. Figure 17 shows even further division of the graph and fewer third party nodes that are being sent information.

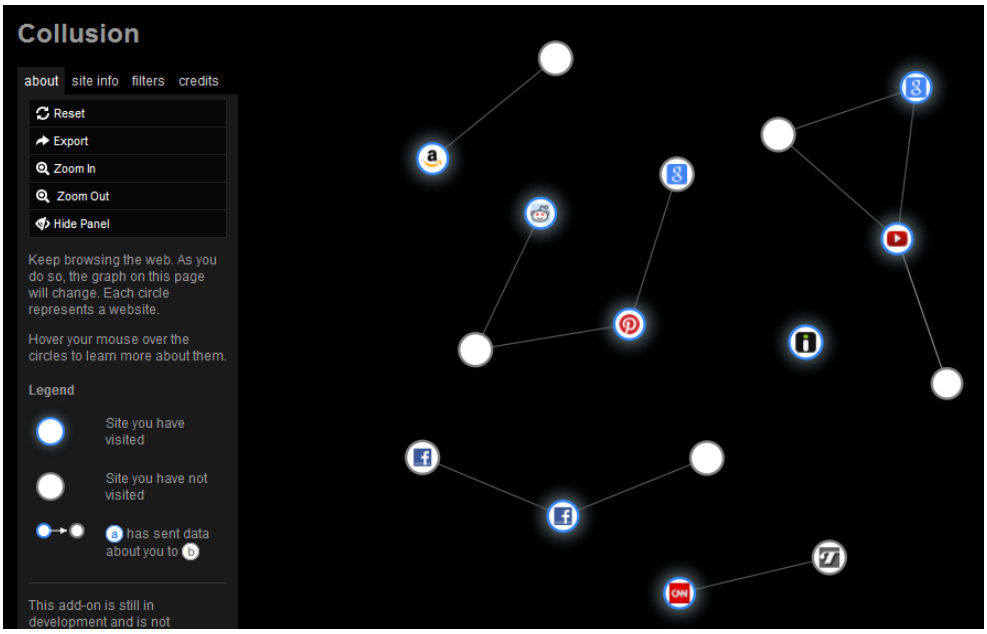


Figure 17: Final Collusion graph with all tools installed

In conclusion, there are multiple ways that private information can be leaked on the Internet. The best way to reclaim privacy online is to be informed. Users are advised to learn how information is transferred on the Internet, how it is tracked and shared, and be aware of how simple things like an IP address can convey a huge amount of personal information. Through the evaluations in this paper it is obvious that there are readily available tools that can help users to reclaim their information privacy online.

References

[1] B. Fitzgerald, *New Copyright Alert System Targets Illegal Downloaders: What Pirates Should Expect,* retrieved from Huffington Post, December 2012, http://www.huffingtonpost.com/2012/10/22/copyright-alert system_n_2003296.html?utm_hp_ref=technology.

[2] S. Vaughan-Nichols, *Big Brother Comes to BitTorrent*, retrieved from *Network World*, December 2012, https://www.networkworld.com/community/node/81658.

[3] MarkMonitor, retrieved December 2012, https://www.markmonitor.com.

[4] *Mark Monitor AntiPiracy: Shut Down the Online Distribution and Promotion of Pirated Digital Content*, retrieved December 2012, https://www.markmonitor.com/download/ds/ds-MarkMonitor_AntiPiracy.pdf.

[5] Privacy violations in social media, retrieved September 2013, http://www.bullguard.com/bullguard-security-center/internet-security/social-media-dangers/privacy-violations-in-social-media.aspx

[6] E. Nakashima and D. Wilber, *Report Says TSA Violated Privacy Law*, retrieved from Washington Post December 2012, http://www.washingtonpost.com/wp-dyn/content/article/2006/12/21/AR2006122101621.html.

[7] A. Roberts, *Store Mannequins May be No Dummies - They're Spies*, retrieved from San Francisco Chronicle, December 2012, http://www.sfgate.com/business/article/Store-mannequins-may-be-no-dummies-they-re-spies-4063639.php.

[8] P. Lobley, *Black Friday Shopper Beware: 'Big Brother' Mannequins are Watching*, retrieved from New Jersey News Room, December 2012, http://www.newjerseynewsroom.com/science-updates/black-friday-shoppers-beware-big-brother-mannequins-are-watching.

[9] B. Guzel, *HTTP Headers for Dummies*, retrieved from *Net Tuts,* December 2012, http://net.tutsplus.com/tutorials/other/http-headers-for-dummies.

[10] *How SSL Works, Geocerts SSL*, retrieved December 2012, http://www.geocerts.com/ssl/how_ssl_works.

[11] L. Story, How *Do They Track You? Let Us Count the Ways*, retrieved from The New York Times, December 2012, http://bits.blogs.nytimes.com/2008/03/09/how-do-they-track-you-let-us-count-the-ways.

[12] M. Brain, *How Internet Cookies Work*, retrieved from *How Stuff Works*, December 2012, http://www.howstuffworks.com/cookie.htm.

[13] R. Smith, *The Web Bug FAQ*, retrieved from Electronic Frontier Foundation, December 2012, https://w2.eff.org/Privacy/Marketing/web_bug.html.

[14] S. Mittal, *User Privacy and the Evolution of Third-party Tracking Mechanisms on the World Wide Web*, retrieved December 2012, http://www.stanford.edu/~sonalm/Mittal_Thesis.pdf.

[15] *Ghostery*, retrieved December 2012, https://www.ghostery.com.

[16] R. Bilton, *Ghostery: A Web Tracking Blocker that Actually Helps the Ad Industry* retrieved from Venture Beat, December 2012, http://venturebeat.com/2012/07/31/ghostery-a-web-tracking-blocker-that-actually-helps-the-ad-industry.

[17] *Evidon: Global Tracker Report, Powered by Ghostery's Visibility over 8M Worldwide Domains*, retrieved from *Business Wire*, December 2012, http://www.businesswire.com/news/home/20120314006134/en/Evidon-Launches-Global-Tracker-Report-Powered-Ghostery'shttps://www.ghostery.com.

[18] *About Tor*, retrieved December 2012, https://www.torproject.org/about/overview.

[19] J, Feigenbaum, Joan, A. Johnson, and et al., *A Model of Onion Routing with Provable Anonymity*, retrieved December 2012, http://www.cs.yale.edu/homes/jf/FJS.pdf.

[20] P. Allani, P. Arora, and et al., *Comparison of VPN Protocols: PPTP*, *IPSEC, L2TP*, retrieved December 2012, http://teal.gmu.edu/courses/ECE543/project/specs-F01/AlArVe.PDF.

[21] *Best VPN Protocol | All VPN Protocols Explained and Compared*, retrieved from WWW December 2012, http://www.bestvpnservice.com/blog/best-vpn-protocol.

Appendix A

1. Firefox version 16.0.2: https://www.mozilla.org/en-US/firefox/fx/

2. HTTPS-Everywhere (Firefox Add-on) version 3: https://www.eff.org/https-everywhere

3. Masking Agent (Firefox Add-on) version 1.2.0: https://addons.mozilla.org/en-US/firefox/addon/masking-agent/?src=search

4. Firebug (Firefox Add-on) version 1.10.6: https://addons.mozilla.org/en-us/firefox/addon/firebug/

5. Tamper Data (Firefox Add-on) version 11.0.1: https://addons.mozilla.org/en-us/firefox/addon/tamper-data/

6. Ghostery (Firefox Add-on) version 2.8.3: https://addons.mozilla.org/en-us/firefox/addon/ghostery/

7. BetterPrivacy (Firefox Add-on) version 1.69: https://addons.mozilla.org/en-us/firefox/addon/betterprivacy/?src=search

8. Collusion (Firefox Add-on) version 0.24: https://addons.mozilla.org/en-us/firefox/addon/collusion/?src=search

9. SQLite Database Browser: http://sourceforge.net/projects/sqlitebrowser/

10. Wireshark version 1.8.3: https://www.wireshark.org/

11. Java version 7 update 9: https://www.java.com/en/download/index.jsp

12. Adobe Flash Player version 11.5.502.110: https://get.adobe.com/flashplayer/

13. TorGuard: http://torguard.net/