APPROVAL SHEET

Title of Thesis: Characteristics of HTML in the Deep Web

Name of Candidate: Matthew Eckert
Master of Science, 2015

Thesis and Abstract Approved: _____

Charles K. Nicholas
Professor
Department of Computer Science and Electrical Engineering

Date Approved: _____

ABSTRACT

Title of Document:	Characteristics of HTML in the Deep Web
	Matthew Eckert
Directed By:	Charles K. Nicholas Professor Department of Computer Science and Electrical Engineering

This paper explores the HTML characteristics of the deep web by gathering HTML tag frequencies on web pages using three different web crawling techniques. The first web crawling technique used the most popular websites listed by Alexa as the seed for the web crawler and randomly selected a sample of web pages to include in the statistics. The second web crawling technique consisted of web pages gathered from randomly generating shorten URLs and visiting pages that the shortened URLs redirected to. The third web crawling technique traversed the deep web going through .onion web sites and domains by randomly generating a IP. Statistics from these web crawling techniques are gathered and compared in this paper.

Characteristics of HTML in the Deep Web

By

Matthew Eckert

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County, in partial fulfillment
of the requirements for the degree of
Master of Science
2015

Table of Contents

List of Tables

Chapter 1: Introduction

The World Wide Web (WWW) is comprised of documents called web pages which allow information to be shared, consumed, created, searched and collaborated on. The World Wide Web is the home of many multi-million dollar companies covering different markets like Google, Facebook, and Pandora just to name a few, sites geared to interest groups, or personal web pages. Some of these companies like Google, Yahoo and Bing make a business of indexing the web to allow users to search for content on the web by crawling web pages.

The World Wide Web can be divided into two components, the surface web and the deep web. This paper will explore three different web crawling techniques to examine usage patterns of HyperText Markup Language (HTML) tags, which define the structure of web pages, over web pages hosted both on the surface and the deep web. This will help us understand if tags are being used the same way throughout the whole Internet and what HTML tags are being used. While examining this we will also look what it takes to index the deep web and what is being done in order to get access to the content in the deep web.

1.1 Why do we care about the World Wide Web?

The Internet is the home of many different types of content and information. It is multi-purposed and what can be done on it is only loosely constrained. The Internet is home to multi-million dollar companies like Google, Facebook and Pandora who make a business of selling ads to visitors of their sites. You have sites that sell goods and services like eBay, Amazon or Newegg. The Internet is very large and constantly changing but its actual size is not known.

1.2 Understanding the Internet- ZMap

ZMap was the first tool used at the start of this research to understand the components of the Internet and to get an understanding of it. “ZMap is an open-source network scanner that enables researchers to easily perform Internet-wide network studies. With a single machine and a well provisioned network uplink, ZMap is capable of performing a complete scan of the IPv4 address space in under five minutes, approaching the theoretical limit of ten gigabit Ethernet. ZMap can be used to study protocol adoption over time, monitor service availability, and help us better understand large systems distributed across the Internet." As stated there are many things you can do with ZMap, and various studies have been conducted such as analysis of HTTPS certificates, understanding network scanning activity, uncovering botnets and identifying and understanding the scope of Heartbleed vulnerabilities. The Heartbleed Bug is a vulnerability in the OpenSSL cryptographic library that allows attackers to invisibly read sensitive data from a web server. This potentially includes cryptographic keys, usernames, and passwords. ZMap was able to find evidence to show that approximately 4.9% of all hosts that support HTTPS remain vulnerable. 6.0% support Heartbleed messages, but are not vulnerable, and 89.1% of HTTPS hosts do not support heartbeat. In total, approximately 1.4 million web servers remained vulnerable two days after the vulnerability being released. ZMap helps us understand many different aspects of the Internet but it does not allow us to examine characteristics of the HTML documents.

1.3 What is HTML?

HTML is the standard markup language used to render a web page. It defines the web page's content, as well as how the page looks and works by defining such things as the layout, font, color and graphics of the web page. Currently 116 valid HTML tags exist in both HTML4 and HTML5.

Different versions of HTML currently exist, with HTML4 and HTML5 being the most recent versions. There are several differences in the two formats, such as ability to extract user location, creating better storage on the client side browser, and built in support for more multimedia functions. Some differences between HTML4 and 5 are: HTML4 contained an <applet> tag that was used for displaying applets in a web browser. However, in HTML5, this applet tag has been removed. In order to display applet type items, a new <object> tag has been introduced in HTML5. Next, HTML4 contained an <acronym> tag that was used for displaying abbreviations in a web browser. However, in HTML5, this tag has been removed. A new <abbr> tag has been introduced in HTML5. Third, the <hr> tag was used to draw a line in HTML4 and all the previous versions of HTML, however in HTML5, the functionality of this tag has been changed and it is used for defining a thematic break in the web page. Forth, In HTML4 and previous versions, the <a> tag was used for both ends of a hyperlink, namely as anchor as well as for referring to a link. In HTML5, the <a> tag is used only as a hyperlink. But if the href attribute is removed from the <a> tag, the <a> tag can be used as a placeholder for other hyperlinks. Lastly, the <meta> tag is defined in the header section of the HTML document and contains information about the data. In the previous versions of HTML, including HTML4, this tag used to

contain an attribute called schema that defined the schema of the document. However, in HTML5, this tag has been removed.

In HTML5 several new tags were added. First, the <canvas> tag which is used to draw graphics using JavaScript. New media tags were introduced such as <audio>, <imbed>, <source>, <track> and <video>. The <mark> tag was also added, which allows the user to highlight text[HTML, 2015].

1.4 What Is Web Crawling?

A web crawler is a bot that programmatically traverses the World Wide Web by starting with a seed of Uniform Resource Locators (URLs) and gathering more URLs from the pages it visits. There are several reasons why web crawlers have been developed. The first and main reason why web crawlers were developed was because of search engines. Crawlers are how a search engine discovers and indexes the content on the World Wide Web. Many different facets of web crawlers have been studied, like selection policy, revisit policy, politeness policy and parallel policy. Selection policy refers to whether a web page should be included in the crawl or not. This policy exists because the Internet is massive and sometimes resources are constrained and not every single page can be visited. When a web page will be re-indexed to retrieve the most up to date content is referred to as the revisit policy. The politeness policy refers to which and how often web pages are visited for one given domain. Since the Internet is big, the parallel policy for a web crawler helps to define how web crawlers crawl using multiple different resources. The architecture of web crawlers has also been well studied. The high level architecture of a web crawler consists of a queue of URLs that need to be referred to a scheduler that sends that

URL to one process that downloads the content, stores the data, and adds new links to the queue. Which technologies to use during each stage could be different based off of the use case.

1.5 Deep Web vs Surface Web

The surface web is the portion of the World Wide Web that is accessible via web crawlers. Web crawlers work by starting out with a set of seed URLs. Seed URLs are usually most effective when they contain pages that have a high amount of links that point outside of their own domain. For each URL, starting with the seed URLs, each of those web pages is visited. When the web crawler visits each page, it extracts the links from the page and then puts those in a queue to be crawled and this process is continued until no more URLs exist in the queue. The surface web content is readily accessible to users that are able to connect to the World Wide Web by websites like Google and Bing.

The deep web refers to the portion of the World Wide Web that is not accessible to web crawlers. When web pages are not accessible via web crawlers, the content cannot easily be indexed in order to be able to be discovered. There are many different reasons why a web crawler would not be able to get to content on the World Wide Web. The first category of content not able to be reached by a web crawler is dynamic web pages that change based on user input. An example of this type of web page would allow content of a database to be searched, such as an academic paper web site, which may only link to papers after they are searched on. Because crawlers navigate the web based off of links that it found on other web pages, web pages that are never linked to are part of the deep web. If a web page is never linked to a web

crawler would never be able to reach the respective web page. Another category belongs to websites that use JavaScript or Ajax or similar technologies because the content dynamically changes based off of how the human interacts with the web page. Web sites that require a login to restrict content are also part of the deep web. For instance, Netflix is a streaming video host that is accessed after paying a subscription. Web crawlers would not get access to Netflix content unless the log in process has been already scripted and paid for. The next category of content that is not able to be indexed is web domains which purposely block content from web crawlers. The most common way to block a web crawler is through using CAPTCHAs which is a challenge presented to a user to test to see if the user is a human or not. The next common form to block web crawlers is by including a Robots Exclusion for the domain. The Robots Exclusion Standard tells a web crawler based off of user agent string where it is and is not allowed to crawl.

```
User-agent: *
Disallow:


# too many repeated hits, too quick
User-agent: litefinder
Disallow: /


# Yahoo. too many repeated hits, too quick
User-agent: Slurp
```

Disallow: /   # too many repeated hits, too quick User-agent: Baidu Disallow: /

Table 1: Robots.txt sample file can this table be kept as a whole, on a single page?

The last portion of the deep web is web sites that are hidden behind a wall of some kind, or can only be accessed via software. The most common example of this is .onion websites that are hidden behind The Onion Router (TOR). TOR is used to allow users to anonymously surf the web by hopping through a network of routers. Another example of websites that are hidden behind software is I2P which is another software technology which allows for anonymous web browsing capabilities [Dingledine *et al*., 2004].

1.6 Paper Motivation

There are multiple motives behind why we would want to explore tag distribution from the surface and deep web. The first is to simply understand what tags are used. Next is to help better understand the transition from HTML 4 to HTML 5. Detecting authorship is another significant aspect. You can look at detecting authors that have created two different web pages based off of how tags are used.

Understanding the deep web and how it is similar or not similar to the surface web is also of high importance because of the size of the deep web and the identified and rich content of the deep web. The reason that so much research has gone into access content/indexing the deep web is because the size is believed to be orders of

magnitude bigger than the surface web and is believed to have a wide range of information. To get the actual size of the deep web is difficult because no one has full access to all of it. Insights into portions of the deep web have given an idea as to the size. The largest portion of the deep web is believed to belong to the National Oceanic Atmospheric Administration (NOAA). One of NOAA's functions is to collect and publish climate and weather data. NOAA hosts searchable databases for this data which is contained in the deep web because web crawlers cannot get access to it. Furthermore, the content of the deep web is believed to be of similar content to what is on the surface web. This claim can be made because by examining particular aspects of the deep web showed that it had similar content to the surface web. The following is areas of content that is believed to be in the deep web: publication such as new paper articles, scholarly articles or encyclopedia such as Association for Computing Machinery's publishing of scholarly articles, shopping and auctions such as websites like Amazon and E-Bay, classifieds for news papers or individual hosted classified web pages, libraries, calculators, and job searching websites like Monster just to name a few. Because of the size of the deep web and the fact that the deep web is believed to house similar content of the surface web gaining insight into the deep web would provide value to users searching, utilizing and processing on the World Wide Web [Bergman, 2001].

Many law enforcement agencies also have a big interest in the deep web and you will see later on one government agency working to tackle some of the problem of indexing the deep web below. Law enforcement personnel are interested in the deep web because it is one area where people go to hide illegal activity. Criminals do

this because they cannot easily be found or related to the activities. One example of this is silk road which was a black market web site hidden behind TOR that sold illegal drugs. Silk Roads has been taken down since an FBI investigation but many websites like Silk Roads still remains online in the hidden web(Burns). Many law enforcement agencies are interested in getting access to the deep web to identify illegal activity.

This paper will explore the HTML tag usage frequency in both the surface and deep web. Understanding HTML tags is very important because it is the building blocks of the web. If we are able to fully understand how HTML tags are used and recognize trends and patterns we will better be able to understand the content of the World Wide Web.

Chapter Two will talk about the work that is currently and has been complete in this area. This will help us to better understand the problem set and answer unanswered questions. Chapter Three will then discuses the three different web crawling techniques that were implemented to gather HTML tag data from both the surface and deep web. Chapter Four will talk about the results that were found from the three different web crawling techniques. Chapter Five will finish the paper with a summary of the work and ways to expand on this work.

Chapter 2: Related Work

Studies and research have been conducted and published on how to build a saleable and effective web crawler. Research has also been conducted about HTML tag frequency. One such group that has looked at tag frequencies is Common Crawl. Common Crawl states "Small startups or even individuals can now access high quality crawl data that was previously only available to large search engine corporations." Analytics have been run over the Common Crawl data set to get statistics on tag distribution but this data set is only covering the surface web.

The first attempt at solving the problem of indexing the deep web was by Google, who came up with Sitemaps in mid 2005. Sitemaps are the exact opposite of Robot Exclusion documents because Sitemaps allow you to tell a web crawler which URLs are able to be crawled. An example sitemap is included below. Sitemaps includes information such as the location of content, which allow the web crawler to be seeded with them. How frequently the domain content changes is also an attribute that is defined in the sitemap. This is important when you talk about a concept called freshness with web crawling. Freshness refers to how often is the content updated, and this is important because with web crawling a limited amount of resources are available, and this will allow the web crawler to know how frequently to check for updates to the web pages. The next part of information that is portrayed in the sitemap is the last time that given URL was updated. This again helps web crawlers with limited resources because with this feature a web crawl would not have to retrieve information that it already has. The last aspect of the sitemap is the priority of URLs

within the given domain. This is specified by the content owner and followed by the web crawler. This tells the web crawler where the most important information is, which again comes into play with limited resources for web crawlers [Sitemaps, 2015].

```
<?xml version="1.0" encoding="UTF-8">
<urlset
xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
    <url>
        <loc> http://www.example.com/</loc>
        <lastmod>2014-12-1</lastmod>
        <changefreq>monthly</changefreq>
        <priority>0.8</priority>
     </url>
<urlset>
```

Table 2: Sitemaps file example better to keep whole, on one page- Will make change when all edits are done.

Content specific searches have also been created to access particular information in the deep web. For instance, Google scholar has built an index of academic papers that are normally only accessed by database searches. By examining Google Scholar you can see that they have integrated certain sites within their search engine that do not get indexed in the normal Google web search. For instance, if you find an abstract in Google Scholar and then search for it in the normal Google search, it will not show up at all or be indexed as high because many more documents are

taken into consideration during that search. Another example of a domain specific access to the deep web is TOURCH, which is specifically made to index .onion top level domain or TOR web pages. To do this they have three techniques of getting web content. The first technique is the normal web crawling technique of starting with a set of seed URLs and visiting sites that are connected to each other. The second technique they use to index .onion websites is allow site owners to register their domain with TOURCH so it can be indexed. The last technique they use to index .onion websites is through brute force guessing of URLs. URLS in the .onion domain are 16 character alphanumeric which allows TOURCH to brute force guess domains since they know the structure of the domains. This is not efficient, but because the .onion surface web is relatively small they are forced to use this method.

A DARPA project called Memex is looking at this problem from a law enforcement perspective which was described above. The Memex program describes its work as follows:

> Today's web searches use a centralized, one-size-fits-all approach that searches the Internet with the same set of tools for all queries. While that model has been wildly successful commercially, it does not work well for many government use cases. To help overcome these challenges, DARPA launched the Memex program in September 2014. Memex seeks to develop software that advances online search capabilities far beyond the current state of the art. The goal is to invent better methods for interacting with and sharing information, so users can quickly and thoroughly organize and search subsets of information relevant to their individual interests. Creation of a new domain-

> specific indexing and search paradigm will provide mechanisms for improved content discovery, information extraction, information retrieval, user collaboration, and extension of current search capabilities to the deep web, the dark web, and nontraditional (e.g. multimedia) content.

Also on the Memex web site is the code behind the process they use so you can look at the different techniques that are used to index the deep web. Memex has been able to find out a handfull of techniques of indexing the deep web and are now starting to share their work with the rest of the community[DARPA-Open Catalog, 2010].

There have been a couple of common approaches for accessing the deep web or making the deep web more accessible. The first approach that has been rising in popularity is the concept of the semantic web, which looks to promote a common data format and exchange protocols on the web. According to the World Wide Web Consortium ,: "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries" [Maedche, 2002]. By having a common structure across the web one off web crawlers to gather specific information would not have to be created. The next biggest technique for accessing the deep web was building domain specific searches to access the hidden data. For instance, if TOURCH took an approach of allowing people to be able to register their domain which allowed it to be discovered if it was not linked to. Another approach is to make sure web site developers know how to create a web page that can be easily indexed. Some approaches to make sure your web page gets indexed include: making sure that your site is linked from somewhere else; Include a sitemap for your domain; Making sure that pages are interlinked so a web crawler can

get to all of the content on the domain. The last technique for making content available to available to a web crawler is to remove access restrictions. A web site developer can do this by not requiring a log-in to access information, allowing the Robots Exclusion not to limit any content on the domain, and not having CAPTCHAs. Another technique for trying to uncover the deep web is to perform simple searches on text boxes. This is high on computational resource demand and not efficient. In this technique, if a textbox or input form is found sample data, random words for example, is put in to find out how the page reacts. The web crawler determines if more information is returned and if so continues to try to find more information behind the search boxes.

Since the deep web has yet to be exposed and the deep web is believed to include an abundant amount of useful information there is still much research left to be done. Since a portion of the deep web is hidden behind areas that only can be accessed based off of user interactions (Searching, clicking, hovering, etc) machine learning techniques are looking to be explored to crawl the web. These machine learning techniques would do things such as filling out forms and submitting them, performing searches, activating JavaScript buttons and allowing flash elements to play just to name some examples. One of the main problems mentioned regarding the machine learning technique is scalability. Considering the size of the world wide web doing this for every web page would be very resource intensive. The next research topic that is being contemplated is how to create a scalable solution. Since the content of the deep web is believed to be 400 times the size of the surface web according "Accessing the Deep Web", the computer resources needed to crawl the entire deep

web is much more than what is needed for the surface web(He et al.). Another approach that can be taken to be able to access the deep web is a query time access to the deep web. In this case a middle ware would work by storing the correct syntax needed to query particular data in the deep web and execute this query during the time the user entered the query. One advantage to this is that the source system maintains the index of the documents making it so the query system would not need to index the documents, saving resources on the query system. This idea can also been seen in the work with semantic web. The common format defined by the semantic webwould allow for systems to interact in a seamless matter and the need for a middleware would not be as great. Many different techniques are being explored in order to access the deep web but not one solution has been able to figure out to access the deep web in its entirety [Raghavan et al., 2000].

Chapter Three will explain the three different web crawling techniques that were used in the paper.

Chapter 3: Experiments

Three different web crawling techniques were used to gather HTML tag statistics. For each technique 50,000 web pages were visited to gather statistics. 50,000 web pages is a very small subset of the Internet but we will see that for some HTML tags this provides you enough to understand tag frequency. For all these web crawling techniques robots.txt files were ignored as we had delays in the web crawling and the amount we hit each site was limited. This was done to get insight into one portion of the deep web.

3.1 Web Crawling Technique One

The first web crawling technique aimed at getting HTML tag data off of the surface web. In order to do that was used to gather web HTML statistics used a seed of popular sites and sampled web pages. This approach followed a breadth first crawl which started with a seed of 300 popular websites home pages that were gathered from Alexa.com which is an Amazon company that gathers Internet activity data. The 300 popular websites home pages were added a FIFO queue. For each web page in the queue it was randomly chosen to be included or not included in the HTML statistics. Furthermore only 10 web pages from a domain were included in the HTML statistics.

3.2 Web Crawling Technique Two

The second web crawling technique used URL shortening services. This technique was geared towards getting data URL shortening services provide a URL that is small with respect to the length of the url that redirects to a different url usually of a longer url length. For this web crawling technique tinyurl tr.im and bitly. HTML tag statistics were gathered from the web pages that were on the end of the redirects from randomly generated URLs of the shortened URL services.

3.3 Web Crawling Technique Three

The third web crawling technique crawled was a combination of two different sources of web pages. The first was similar to the first web crawling techniques where it was a breadth first crawl but instead of choosing if a web page should be included or not randomly all web pages were included if ten web pages from the domain had not been visited. Unlike the first technique where the seed URLs are

popular websites the seed urls are .onion sites that were gathered from various lists of .onion domains. Furthermore, for every ten .onion web pages that were crawled ten web pages were tempted to be crawled by randomly generating an IP and checking to see if a website was hosted on that IP. The reason that we are did this web crawling technique is because it is gathering HTML tag data from a portion of the deep web. Chapter 4 will dive into the results of the gathered from using the three web crawling techniques and compare the different result sets.

Chapter 4: Results

4.1 Web Crawling Technique One Results

Several tags were never seen when using the first web crawling technique. These tags include: <applet>, <basefont>, <bdo>, <dialog>, <dir>, <keygen>, <meter>, <output>, <progress>, <rp>, <rt> and <ruby>. The following tags were used the most frequently in html documents: <a>, <p>, <li>, <i>, , <img>, <script>, <ul>, and <td>. This helps to shape our understanding of what would make a web page unique if it used certain tags. Seeing that the <a> tag is the most frequent tag used shows that these sites are highly interlinked making those pages more discoverable. This is expected because in this technique we started with the most popular web sites. Table 3 shows the most popular tags with the max number of times they were seen on a page, the average and standardization of how many times a tag is seen on a web page.

Tag	Min	Max	Avg	Std
<a>	0	632	118.1856	87.7833

<p>	0	398	70.0773	76.8221
	0	500	63.0539	68.1911
<i>	0	254	46.9012	89.6573
	0	407	35.0776	60.5071
<img>	0	409	19.0583	32.9407
<script>	0	354	13.5783	14.4212
	0	265	12.044	19.7599
<td>	0	254	11.7102	37.7768

Table 3: Web Crawling Technique One Statistics

Several tags converged to within ½ plus or minus for more than 500 web pages. Some of the convergences were not interesting as they were seen less than an average of two times seen on a web page. For instance, the title tag was on average seen 1.09 which makes sense because the majority of the time a web page has one title unless they web page holds multiple different pieces of information and changes based off of selection. Other tags converged because the amount of times that they were included in an html document were small. For instance, h6 was seen in 3% of html documents collected. Table four show to ten tags out of twenty one were seen in less than 5% of the HTML documents excluding the ones that were never seen.

Tag	**Seen in Percentage**
<track>	.05
<var>	.06
<mark>	.21
<menuitem>	.35

<audio>	.46
<noframes>	.76
<datalist>	.79
<frameset>	.92
<details>	.97
<bdi>	1.01
<kbd>	1.2

Table 4: Seen in percentage for Web Crawling technique one

Several tags that were seen in more than 25% of the HTML documents did converge and you can examples of that in the see that in the <ul>, <small>, <scrpit> and <html> tags. The <ul> and <script> tag both converged after visiting around 45,000 at approximately 12.5 and 13.5 respectively. Less interesting <html> converged at an average of one per page along with the <small> tag.

Standard deviation tells us the amount of variation or dispersion of the data set. Table six shows the highest standard deviations in the data while Table five shows the tags with the smallest deviation excluding tags that were seen in less than 5% of documents.

Tag	**Standard Deviation**
<track>	0.0399
<mark>	0.0411
<frameset>	0.0488
<tfoot>	0.0717
<caption>	0.0879

<details>	0.1465
<q>	0.1505
<canvas>	0.1534
<big>	0.1631
<frame>	0.1656
<html>	0.1926

Table 5: Low standard deviation for technique one

Tag	**Standard Deviation**
<i>	89.6573
<a>	87.7833
<p>	76.8221
	68.1911
	60.5071
<td>	37.7768
<b>	33.8788
<img>	32.9407
 	29.2822
<tr>	26.8981
	19.7599

Table 6: High standard deviation for technique one

4.2 Web Crawling Technique Two Results

Like web crawling technique one several tags were never seen when using the second web crawling technique. These tags include: <applet>, <track>, <mark>, <frame>, <keygen>, <datalist>, <noframe>, <rp>, <rt> and <ruby>. The following tags were used the most frequently in html documents: <a>, <li>, <i>, <img>, , <script>,
, <ul>, <meta>, and <link>. Table seven, shows the minimum, maximum, average, and standard deviation for web crawling technique two most popular tags based off of average.

Tag	Min	Max	Avg	Std
<a>	0	436	143.5269	91.6372
	0	381	97.2342	82.2673
<img>	0	431	92.1529	81.8246
	0	457	41.843	74.3284
<script>	0	502	37.2753	63.3645
 	0	397	22.0473	47.6247
	0	398	19.8459	41.7744
<meta>	0	303	16.1947	35.1045
<link>	0	298	12.4723	31.4583

Table 7:Tag statistics for technique two

Several tags average converged to within ½ plus or minus for more than 500 web pages. Again we removed the tags that were seen in less than five percent of web pages. For instance, <h6> was seen in 3% of html documents collected so this was

removed. Table eight shows the top 10 tags of seventeen were seen in less than 5% of the HTML documents excluding the ones that were never seen.

Tag	Seen in Percentage
<colgroup>	.03
<big>	.05
<canvas>	.15
<strike>	.20
<samp>	.29
<frame>	.49
<wbr>	.53
<caption>	.78
<q>	.86
<tfoot>	.95
<kbd>	1.06

Table 8: Seen in percentage for technique two

Several tags averages converged that were seen in more than 25% of the HTML documents did converge in these technique as well. The <var> tag converged at an average of nineteen after around 25,000 web pages were visited. Similarly, the <frameset> tag converged after about 15,000 tags at an average of around three. Table Ten shows the highest standard deviations in the data while Table Nine shows the tags with the smallest deviation excluding tags that were seen in less than five percent of documents.

Tag	Standard Deviation

<main>	0.0413
<menuitem>	0.0494
<frameset>	0.0636
<mark>	0.0698
<q>	0.0785
<bdi>	0.0894
<var>	0.1002
<menu>	0.103
<optgroup>	0.1043
<acronym>	0.1064
<col>	0.1086

Table 9: low standard deviation for technique two

Tag	**Standard Deviation**
<p>	91.4286
<a>	84.4575
<i>	75.7352
<meta>	73.6527
<link>	60.7246
	53.5728
<img>	39.4347
<h3>	34.2854
<head>	30.8475

	27.3627
<tr>	22.4727

Table 10: High standard deviation for technique two

4.3 Web Crawling Technique Thee Results

Like web crawling technique one several tags were never seen when using the second web crawling technique. These tags include: applet, <basefont>, <bdo>, <dialog>, <dir>, <keygen>, <meter>, <output>, <progress>, <rp>, <rt> and <ruby>. The following tags were used the most frequently in html documents: <a>, <li>, <i>, <img>, , <script>,
, <ul>, <meta>, and <link>. Table eleven shows the minimum, maximum, average, and standard deviation for web crawling technique two most popular tags based off of average.

Tag	**Min**	**Max**	**Avg**	**Std**
<a>	0	543	143.0032	83.
<p>	0	501	111.1723	93.
	0	459	102.9247	63.
<i>	0	479	87.3960	67.
	0	502	62.0437	37.
<img>	0	405	52.3271	47.
<script>	0	329	51.3954	41.
	0	387	43.2734	35.
<td>	0	302	37.5829	31.

Table 11:Tag statistics for technique three

Several tags average converged to within ½ plus or minus for more than 500 web pages. Again we removed the tags that were seen in less than five percent of web pages. Table twelve shows the top 10 tags of seventeen were seen in less than 5% of the HTML documents excluding the ones that were never seen.

Tag	**Seen in Percentage**
<menuitem>	.09
<main>	.17
<bdi>	.17
<tfoot>	.23
<details>	.31
<strike>	.54
<noframes>	.74
<kbd>	1.02
<audio>	1.24
<wbr>	1.35
<menuitem>	.09

Table 12: Seen in percentage for technique three

Several tags averages converged that were seen in more than 25% of the HTML documents did converge in these technique as well. The <button> tag converged at an average of five after around 37,000 web pages were visited. Similarly, the <hr> tag converged after about 21,500 tags at an average of around seven and a half. Also, the <h3> tag converged at eleven and a half at around 33,500 web pages visited.

Table thirteen shows the highest standard deviations in the data while Table fourteen shows the tags with the smallest deviation excluding tags that were seen in less than five percent of documents.

Tag	Standard Deviation
<video>	0.0853
<hr>	0.0914
<button>	0.1053
<dt>	0.1130
<samp>	0.1198
<frameset>	0.1353
<q>	0.1427
<h4>	0.1438
<abbr>	0.1490
<video>	0.0853
<hr>	0.0914

Table 13: low standard deviation for technique three

Tag	Standard Deviation
<i>	102.4799
<p>	89.2001
<a>	78.5812
	68.2557
	67.2854

	59.3725
<img>	53.2385
<td>	47.2752
<tr>	39.4927
 	37.6826
<i>	102.4799

Table 14: High standard deviation for technique three

4.4 Comparing Web Crawling Techniques

Several tags were never seen in the three different web crawling techniques. These tags are: <applet>, <rt>, <rp>, <ruby> and <keygen>. Of the other tags that were not seen all were seen in less than five percent of the other web crawling techniques. This tells us that the tags that are not frequently used or never seen are the same through three different web crawling techniques. Similarly tags that are used more frequently were the same through the three different web crawling techniques. Some of these tags were: <a>, <i>, <li>, <img>, <ul> and <script>. Some tags also had a high standard deviation and others had a low standard deviation. What the standard deviation tells us is how much variation there is in the tag. Tags that were more popular tended to have a larger standard deviation then tags that were seen less, filtering out tags that were in less than five percent of the documents. Tags that had a low standard deviation were not the same throughout the three web crawling techniques. Tags with a high standard deviation did have similar tags through the three web crawling techniques like: <i>, <a>, <p>,<img> and <tr>.

There were also some patterns in that would help predict if a tag would converge or not. Tags with low standard deviations converged more frequently then tags with higher standard deviations. Also, tags that were seen in a lower percentage of web pages converged more frequently. Tags that had a low average, below ten, also converged more frequently.

The biggest variations distinction in tag usage came with the second web crawling technique which involved shortened URL services. The variation came into play as more "content" related tags had a higher frequency of usage. This would be tags like <img>, <video>, <audio>, <imbed>, <source>. One reason behind this behavior is that we shortened URL services are predominantly used to share content from one user to the next.

With this data you can also have some insight on web pages shifting from HTML4 to HTML5. All three web crawling techniques saw the use of new HTML5 tags. The first web crawling technique where we started with popular sites as seeds and shortened URL technique saw more HTML5 tags then the third web crawling technique which focused on the deep web. This could be because the surface web is more adaptive to following the norms and setting the structure of the web.

Furthermore, some similar tag distributions were seen in public Common Crawl result sets that analyze tag distribution. The actually comparisons between the two different data sets was hard because different forms of output were produced. Common Crawl public analytics outputted the total count of the tags seen in the data set. This allowed us to see what percentage of the total tags a tag was but not information like average or standard deviation of a tag on a web page.

4.5 Malware Tag Frequency

Now that the tag distribution is known for three different web crawling techniques we can compare those tag frequencies to other data sets. In this example, we looked to see if HTML documents that have been identified as having malware has a unique tag distribution to help to identify potential HTML documents containing malware. A sample of identified malware Packet Captures(PCAPs), network traffic captured into a text file, files were extracted to get the HTML files. Tag frequencies were extracted from these documents using the process above to analyze the three different web crawling techniques. The PCAP samples of malware was to small to come up with a conclusive evidence as to tag distribution as the data set was small and no similar convergence was evident like in the above three techniques. Convergence occurred in small pockets but the sample size was much smaller than the above techniques so it is inconclusive if the malware HTML files have a different tag distribution.

Chapter five will summarize the work in this paper and purpose follow on work to be completed.

Chapter 5 Conclusion

Three web crawling techniques were used to extract HTML tag frequencies to see if there is any variation in how tags are used in different portions of the web. The first web crawling technique started with a seed of the top web sites on the World Wide Web and the selection process of if a web page's tag statistics were to be included were if the domain was not included in the tag statistics up to ten times and that the page was randomly decided to be included or not included. The second web

crawling technique had the same constraint of if a page was included or not included but the key difference is the second web crawling technique all web pages were linked to by a shorten URL service. The last web crawling technique focused on crawling the deep web particularly the portions of the deep web hidden behind the TOR hidden services.

5.1 Future Work

The above work can be expanded in multiple ways. First, gathering more HTML documents to see if that makes a difference on the tag distribution. Resources were limited in this research by machine size and network bandwidth. This web crawlers were not optimized or parallelized but could be to increase the effectiveness of data acquisition. Papers like “Effective Web Crawling” by Carlos Castillo and “High Performance Web Crawling” by Marc Najork and Allan Heydon describe how to create more optimized web crawlers. The next area of expansion is to look how HTML tag attributes are used. Attributes are a key value pair with the keys primarily being id, title, class, and style which define information about the HTML tag. For example, you could have <img scr=“example.jpg” id=“example-img” width=“100” height= “150”> where <img? is the tag and the attributes are scr, id, width, height. The reason we would want to look at how attributes are utilized is because this could help further identify authorship. With this data set and knowledge comparisons of different data acquisitions for instance the malware data set or another web crawling technique can be compared to the above data set. This will help us to understanding different portions of the World Wide Web. One last additional way that this work can be expanded is by including in more of the deep web crawling techniques. This would

help to get a more accurate statistics from the deep web. As we have learned the deep web is massive so we would need to crawl the deep web in different position to get an accurate sample of the deep web. Machine learning techniques, like SVM, K Nearest Neighbor, K Means Clustering and many others can now be used to used to see if you can cluster together web pages based off of tag statistics.

References

[Bergman, 2001] Bergman, Michael K. "White paper: the deep web: surfacing hidden value."Journal of Electronic publishing 7.1 (2001).

[Berners-Lee et al., 2001] Berners-Lee, Tim, James Hendler, and Ora Lassila. "The semantic web."Scientific American 284.5 (2001): 28-37.

[Burns, 2014] Burns, Matt. "FBI Seizes Deep Web Black Market Silk Road, Arrests Owner". TechCrunch. Retrieved 8 February 2014.

[Castillo, 2005] Castillo, Carlos. "Effective web crawling." ACM SIGIR Forum. Vol. 39. No. 1. ACM, 2005.

[TOR, 2015] "TOR." Project: Anonymity Online.N.p., n.d. Web 05 May 2015

[Chakrabarti et al., 2002] Chakrabarti, Soumen, Byron Edward Dom, and Martin Henk van den Berg. "System and method for focussed web crawling." U.S. Patent No. 6,418,433. 9 Jul. 2002.

[Common Crawl, 2015] "Common Crawl." Common Crawl.N.p., n.d. Web 03 Oct. 2015

[DARPA-Open Catalog, 2010] "DARPA- Open Catalog." DARPA - Open Catalog. N.p., n.d. Web. 05 May 2015.

[Neumann et al., 2010] Neumann, Alexander, Johannes Barnickel, and Ulrike Meyer. "Security and privacy implications of url shortening services." Proceedings of the Workshop on Web 2.0 Security and Privacy. 2010.

[Dingledine *et al*., 2004] Dingledine, Roger, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Naval Research Lab Washington DC, 2004.

[Elsas et al., 2004] Elsas, Jonathan, and Miles Efron. "HTML tag based metrics for use in web page type classification." American Society for Information Science and Technology Annual Meeting. 2004.

[Gupta et al., 2003] Gupta, Suhit, et al. "DOM-based content extraction of HTML documents."Proceedings of the 12th international conference on World Wide Web. ACM, 2003.

[He et al., 2007] He, Bin, et al. "Accessing the deep web." Communications of the ACM 50.5 (2007): 94-101.

[HTML, 2015] "HTML4 vs HTML5 Comparison." Go4ExpertWeb Design HTML and CSS Tutorials RSS. N.p., n.d. Web 20 Sept 2015.

[Lawrence et al., 1999] Lawrence, Steve, and C. Lee Giles. "Accessibility of information on the web."Nature 400.6740 (1999): 107-107.

[Liu et al., 2003] Liu, Bing, Robert Grossman, and Yanhong Zhai. "Mining data records in Web pages." Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2003.

[Madhava et al., 2008] Madhavan, Jayant, et al. "Google's deep web crawl." Proceedings of the VLDB Endowment 1.2 (2008): 1241-1252.

[Madhava et al., Harnnesing, 2009] Madhavan, Jayant, et al. "Harnessing the deep web: Present and future." arXiv preprint arXiv:0909.1785 (2009).

[Maedche, 2002] Maedche, Alexander. Ontology learning for the semantic web. Springer Science & Business Media, 2002.

[Myllymaki et al., 2002] Myllymaki, Jussi. "Effective web data extraction with standard XML technologies." Computer Networks 39.5 (2002): 635-644.

[Najork et al., 2002] Najork, Marc, and Allan Heydon. High-performance web crawling. Springer US, 2002

[Peisu et al., 2008] Peisu, Xiang, Tian Ke, and Huang Qinzhen. "A framework of deep Web crawler." Control Conference, 2008. CCC 2008. 27th Chinese. IEEE, 2008.

[Raghavan et al., 2000] Raghavan, Sriram, and Hector Garcia-Molina. "Crawling the hidden web." (2000).

[Ru et al., 2005] Ru, Yanbo, and Ellis Horowitz. "Indexing the invisible web: a survey." Online Information Review 29.3 (2005): 249-265

[Singh, 2002] Singh, Munindar P. "Deep web structure." Internet Computing, IEEE 6.5 (2002): 4-5.

[Sitemaps, 2015] "Sitemaps.org." Sitemaps.org -Protocol.N.p., n.d. Web. 05 May 2015.