The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
21-23 September 2017, Bucharest, Romania

The Problem of Interceptor Top Level Domains Revisited – Extended Abstract

George Markowsky[1]
[1] Cybersecurity Lab, University of Maine, Orono, ME 04469-5711,markov@maine.edu,cybermaine.org

Abstract – **This paper is an extension of my earlier paper [1]. It examines the continuing growth of this problem, and the likelihood that it will continue to get worse. Since the previous paper, the author was contacted by a lawyer involved in litigating a dispute related to TLDs. As noted in the previous paper, misdirected communication has obvious security and privacy ramifications and is an age old problem that predates the Internet. In recent years it has been exacerbated by the arrival of electronic communication and the multitude of ways that communication can be misdirected. A new chapter in this age old problem is being written with the proliferation of top level domains (TLDs). This paper examines this problem in more detail and provides a full analysis of the current set of TLDs. In particular, we are concerned with TLDs that have a non-trivial probability of intercepting traffic intended for another TLD. Intercepting TLDs have the potential of intercepting a lot of traffic since a TLD can support many domain names. This paper updates the results in the previous paper and also performs an auditory similarity measurement between domain names. The issue here is that sometimes TLDs are given orally, such as over the telephone. There are TLDs which, although spelled differently, sound similar to each other. Finally, we give the results of some experiments performed with purchased domain names to be deliberately confusing.**

Keywords – **domain interception, TLD, auto-completion, Demerau-Levenshtein distance, prefix chain, suffix tree, restricted prefix order**

I. Introduction

Communication going to the wrong address is an age old problem. Most are familiar with misaddressed letters and wrong numbers. A misaddressed letter is part of the plot of the Hercule Poirot mystery *The A.B.C. Murders* [2]. With some simpler forms of communication, interception was not easy to do since one could not easily cause mail to be misaddressed or phone numbers to be misdialed. A recent high profile example of misaddressed email involved the firms Goldman Sachs and Google [3]: a Goldman Sachs contractor mistakenly emailed documents intended for x@gs.com to x@gmail.com.

With the advent of the Internet and domain names people were able to agressively pursue strategies for intercepting communications intended for others. Intentionally trying to register domains so as to misdirect traffic is called "cybersquatting" [4] and has been the cause of legislation and litigation. A particular form of cybersquatting is known as "typosquatting" [5] and is also very common.

According to Wikipedia [4], "cybersquatting is registering, trafficking in, or using an Internet domain with bad faith. There is an extensive legal and technical literature on the subject of cybersquatting. References [6], [7] and [8] provide many additional details and examples dealing with the problem of domain name confusion within the context of a single TLD. They also address these issues from a financial perspective. The papers by Badgley [9] and Gilfoil [10] discuss some of the legislation that has been passed to help deal with the cybersquatting problem. The paper by Chen [11] gives some interesting background on the approval of TLDs by ICANN. It also notes that completely protecting domains by registering many common variants of 20 trademarks in the various TLDs might cost something like $10,344,800 per year, which is a sum that is significant for most entities. None of the papers in the references address the problem of interceptor domains that is defined and analyzed in this paper.

We say that one TLD (top level domain), A, is an interceptor domain for another TLD, B, if there is a non-trivial probability that traffic intended for B might end up at A. It is possible for two TLDs to each intercept the other. Consider the example of the two TLDs .career and .careers. It does not take much imagination to realize that these two TLDs are likely to be confused with each other and that traffic intended for one might go to the other with serious security and privacy concerns.

There are three broad classes of errors that can cause one domain to intercept another. First, errors caused by auto-completion either by extending a domain name or failing to extend it. Second, typing errors of various sorts. These include transpositions of letters, missing letters or extra letters. Third, pairs of domain names that might be confusing for various reasons. This includes pairs such as .engineer and .engineering in addition to the .career and .careers pair already mentioned.

As of July 10, 2015 there were 1006 TLDs available. There are perhaps another 1,000 on the way to being registered. Because of these numbers it is not possible to give all details about all domains in this paper. My website (http://DrGM.us) will provide frequent reports that will

Date	Number of TLDs
July 11, 2014	639
February 28, 2015	831
July 10, 2015	1,006
September 19, 2015	1,061
October 27, 2015	1,082
September 19, 2016	1,494
March 18, 2017	1,530

Figure 1. The Number of TLDs over Time

contain complete information about each TLD available at the time the report was compiled. Of the 1006 TLDs analyzed for this paper, we find that at least 772 (roughly 77%) are at risk for interception by another TLD. The rapid growth in the number of TLDs is shown in Figure 1.

An *interceptor TLD* is a TLD that is "similar enough" to another TLD that it can capture traffic intended for the other TLD. Often the relation of interceptor is symmetric because the relations "similar enough" is symmetric. Capturing traffic can take the form of putting up alterntive websites or setting up "catch-all" email addresses. A catch-all email address is the address to which all email that is not directed to a real email address at that domain goes. If one sets up only one email address for a domain and makes it a catch-all address, then it will capture all email that comes to that domain.

When one TLD (top level domain) name is a prefix or suffix of another it is possible for one TLD to intercept traffic intended for the other. We analyze domain interception by studying the prefix ordering between TLDs. We restrict this prefix ordering to make it more useful in the study of TLDs. In Section II we show that the restricted prefix ordering has many of the same properties as the standard prefix ordering. This restricted prefix ordering permits us to partition the set of TLDs into maximal suffix trees which permit reasonably efficient calculation of various sorts of interception.

For example, as noted earlier the TLDs .career and .careers are interceptor domains for each other. They differ by just one character and it is easy to imagine that people will be careless about remembering whether .career or .careers is the proper TLD. Suppose that somene has registered the domain good.career and has an email address of smith@good.career. Suppose a different person has registered good.careers. If that person sets up the email address jones@good.careers as a catch-all email address then all email addressed to anyone at good.careers will come to jones@good.careers. It is not necessary for the owner of jones@good.careers to know the possible email addresses at good.career since any email address with s added to the end of .career will go to jones@good.careers. In particular, an email addressed to smith@good.careers will go to jones@good.careers. Similarly, one can shadow the website at good.career and make it so that the casual user does not notice that the wrong web page is being visited. It is clear that domain interception can lead to severe security and privacy compromises.

This paper describes some of the combinatorial features of the TLD space and programs that help produce the results in this paper. We believe that the information in this paper can help people avoid problems when selecting a domain name. Additionally, it will make them aware of domains that might intercept the domains that they already have. We also hope that it might influence people seeking to set up new domains to pick names that are less likely to intercept or be intercepted by TLDs already chosen. Of particular interest are the TLDs known as country codes. Since they are all just two letters long it is clear that there is a lot of potential for confusion. We discuss the problem of country codes in Section III. We also hope that the results will enable domain name registrants to better understand the risks involved with choices of particular TLDs. The results in this paper also suggest that the current system of accepting new TLDs be improved so that new TLDs do not exacerbate the problem.

In this paper we work with the Latin letter versions of TLDs and the auditory versions of TLDs. There may be other confusions possible with internationalized country code TLDs, and they will not be considered in this paper.

II. Prefix Ordering and Autocompletion

This material is taken from my earlier paper [1]. This section describes some natural partial orders on the space of finite length strings. These partial orders help us understand the mathematical structure of the space of domain names. For the remainder of this paper, we use A to denote some fixed alphabet. Typically, it is the alphabet of permitted characters in defining some particular class of names such as TLDs. The space of all finite length strings over the alphabet A is denoted by A*. We shall use Q to denote some subset of A*.

A. Prefix Ordering and Autocompletion Errors

To help describe strings we will use Python notation and conventions. This means that all string indices for a string s begin at 0 and run until len(s) - 1 where len is the length of the string. Negative indices indicate indexing that begins at the end of the string. Thus, s[-1] refers to the last character of the string s. Finally, s[1:6] represents the substring of s formed by taking s[1]s[2]...s[5]. In particular, s[:-1] is the substring that results when the last character is dropped. The previous example shows that when arguments are omitted we substitute either 0 or len(s) as appropriate.

A* is totally ordered by the familiar *alphabetical ordering* which we denoted by $\leq_a$. There are many interesting properties of the partial order (A*,$\leq_a$) that we will not have time to discuss in this paper

Definition 1. *For two strings v and w we write that $v \leq_p w$ iff v is a* prefix *of w, or, alternatively, w is a* suffix *of v.* ■

The connection between autocompletion errors and the prefix partial ordering are fairly obvious. If a string, s, is a prefix of another string, t, it is possible to get s while intending to type t simply by not typing all the characters. This occasionally happens when key strokes are not energetic enough to register a character. Alternatively, one can get t while intending to type s by having the text input get "auto-completed" to t. We will turn our attention to more fully understanding the prefix ordering and seeing how it leads to a partition of the TLDs.

Theorem 1. *Let u, v, w be in A*. If $u \leq_a v$, $v \leq_a w$, and $u \leq_p w$, then $u \leq_p v$.*

Proof: If $u \not\leq_p v$, then there is an integer $0 < n <$ len(n) -1 such that $u[:n] \leq_p v$, but $u[:n+1] \not\leq_p v$. This means that u[:n] = v[:n] and $u[:n+1] <_a v[:n+1]$ since $u[:n+1] \not\leq_p v[:n+1]$. However, since u[:n+1] = w[:n+1], we would have $v \not\leq_a w$ which contradicts the assumption that $v \leq_a w$. ■

To get more insight into the TLD intersection problem, we will restrict the prefix ordering to make it more useful in the study of TLDs. More generally, let Q be a subset of A*. We use $\leq_Q$ to represent $\leq_p$ when restricted to elements of Q. In particular when we write $v \leq_Q w$ we imply that both v and w are elements of Q.

Definition 2. *Given a subset Q of A*, we call $\leq_Q$ the restricted prefix ordering on Q.* ■

Note that if Q = A*, $\leq_Q$ is just $\leq_p$.

Theorem 2. *(*The Prefix Theorem*). Given u, v, w in A*, if $u \leq_p w$ and $v \leq_p w$, then either $u \leq_p v$ or $v \leq_p u$. If Q is a subset of A*, we have that if $u \leq_Q w$ and $v \leq_Q w$, then either $u \leq_Q v$ or $v \leq_Q u$.*

Proof: Since $u \leq_p w$ and $v \leq_p w$ it follows that u = w[:i] and v = w[:j] for some integers $i, j \leq$ len(w). If $i \leq j$ then $u \leq_p v$ otherwise $v \leq_p u$.

Note that exactly the same proof holds in the case of $\leq_Q$. ■

Corollary 3. *The poset (partially ordered set) P = (A*,$\leq_p$) is a meet semilattice, i.e., any two elements in A* have a greatest lower bound.*

Proof: Let v and w be any two elements of A*. Since the empty string λ is in A*, we know that v and w have a lower bound so the set of lower bounds of {v, w} is non-empty. By Theorem 2 we know that the set of all lower bounds of {v, w} is linearly ordered. Since it is finite, it has a largest element which is the greatest lower bound of v and w. ■

Note that Corollary 3 does not hold in general for the poset (Q,$\leq_Q$). It holds if λ is in Q, but otherwise it depends the membership of Q. The following corollary holds for arbitrary Q.

Corollary 4. *Let Q be a subset of A*. The poset (partially ordered set) P = (Q,$\leq_Q$) is* coherent*, i.e., any two elements in Q that have a lower bound have a greatest lower bound.*

Proof: This is essentially the same proof as for Corollary 3. Let v and w be any two elements of Q. Since v and w have a lower bound the set of lower bounds of {v, w} is non-empty. By Theorem 2 we know that the set of all lower bounds of {v, w} is linearly ordered. Since it is finite, it has a largest element which is the greatest lower bound of v and w. ■

B. Suffix Trees

So far we have focused on prefixes of strings. The existence of autocompletion errors suggests that we should devote some attention to suffixes. Let $v \in Q$, define $suf_Q(v) = \{w \in Q \mid v \leq_Q w\}$. It follows from the Prefix Theorem (Theorem 2) that $suf_Q(v)$ is a tree, hence the term *suffix tree*.

Note that $suf_{A*}(\lambda)$ = A*, so there is essentially just one maximal suffix tree. For $Q \neq A*$, there are in general multiple suffix trees. It is clear that each minimal element (with respect to $\leq_Q$) in Q is the root of a maximal suffix tree. This motivates the following definition.

Definition 3. *Let Q be an arbitrary subset of A*. Let min_Q = { q in Q | q is minimal }. We call this set the* roots *of Q.* ■

It is clear that the union of the suffix trees of all the roots of Q is exactly Q. On July 10, 2015 TLD had 557 roots.

III. Typing Errors

This section is taken from my earlier paper [1]. In Section II we discussed how domains can be intercepted because they are either prefixes or suffixes of each other. In this section we discuss domains that can be confused because they are "close" to each other. We will study closeness in terms of the Demerau-Levenshtein distance. Besides the Demerau-Levenshtein distance there are a number of other distances that can be defined on A*. We will not pursue these other distances further in this paper. For more information on this topic consult the paper by Navarro [13].

A. The Demerau-Levenshtein Distance

Definition 4. *Given two strings v and w, the* Demerau-Levenshtein distance *[12], denoted* DL(v,w) *is the number of operations it takes to transform string v into string*

w, where the permitted operations are insertion, deletion, substitution for a single character and transposition of adjacent characters. ■

The types of operations mentioned in Definition 4 correspond to the types of errors that people make when inputting text. In fact, understanding the impact of such errors was one of the key motivators for defining the Demerau-Levenshtein distance in the first place. In this paper, we will focus on domains having Demerau-Levenshtein distance = 1. There are two reasons for this. First, Demerau-Levenshtein distance = 1 is more likely to happen than Demerau-Levenshtein distance ≥ 2. Second, as we shall soon see there are extremely many TLDs within Demerau-Levenshtein distance of 2 of each other.

B. Country Codes

There are no single letter TLDs, although there are single-letter domains for some TLDs. Every country is entitled to a two-letter TLD and many have them. Some countries have put residency restrictions on who would be allowed to purchase domains in their TLD. One of the reasons we did not pursue TLDs with a Demerau-Levenshtein distance of 2 or more is that all country codes are just a Demerau-Levenshtein distance of 2 apart from each other. The country specific TLDs can cause interception problems and themselves be intercepted as can be seen from the report that we generate. This report is discussed in more detail in Section 6.

IV. User Confusion

Aside from the interception that can take place because of autocompletion and typos, interception can occur because of confusion on the part of the user. We have already noted that the TLDs .career and .careers are very close typographically. We also suspect that users might be careless in remembering whether the TLD in some particular instance was .career or .careers. Other similarly confusing pairs are .accountant and .accountants, .ad and .ads, and .auto and .autos.

There are pairs of TLDs that are not very close typographically, but which we predict are likely to be confused. These include the pairs (engineer,engineering), (finance,financial), and (fit,fitness). We have not identified all such confusing pairs at this time.

V. Statistics

Figure 2 shows the prefix tree with CA as the root. It contains 30 elements. Figure 3 shows the prefix tree with CO as the root. It contains 28 elements. Finally, 4 shows the prefix tree with RE as the root. It contains 25 elements. Finally, **??**

VI. Classical TLDs

It is of particular interest to see how the "classical" TLDs have fared in the proliferation of TLDs. The following subsections summarize these results.

- CA
  - CAB
  - CAFÉ
  - CAL
    - CALL
  - CALVINKLEIN
  - CAM
    - CAMERA
    - CAMP
  - CANCERRESEARCH
  - CANON
  - CAPETOWN
  - CAPITAL
    - CAPITALONE
  - CAR
    - CARAVAN
    - CARDS
    - CARE
    - CAREER
    - CAREERS
    - CARS
    - CARTIER
  - CASA
  - CASE
  - CASEIH
  - CASH
  - CASINO
  - CAT
    - CATERING
    - CATHOLIC

Figure 2. The CA Prefix Tree

- CO
  - COACH
  - CODES
  - COFFEE
  - COLLEGE
  - COLOGNE
  - COM
    - COMCAST
    - COMMBANK
    - COMMUNITY
    - COMPANY
    - COMPARE
    - COMPUTER
    - COMSEC
  - CONDOS
  - CONSTRUCTION
  - CONSULTING
  - CONTACT
  - CONTRACTORS
  - COOKING
  - COOKINGCHANNEL
  - COOL
  - COOP
  - CORSICA
  - COUNTRY
  - COUPON
    - COUPONS
  - COURSES

Figure 3. The CA Prefix Tree

O

RE

READ
REALESTATE
REALTOR
REALTY
RECIPES
RED
REDSTONE
REDUMBRELLA
REHAB
REISE
REISEN
REIT
RELIANCE
REN
RENT
RENTALS
REPAIR
REPORT
REPUBLICAN
REST
RESTAURANT
REVIEW
REVIEWS
REXROTH

Figure 4. The RE Prefix Tree

Size of Tree	Number of Trees	Size of Tree	Number of Trees
30	1	15	3
29	0	14	2
28	1	13	3
27	0	12	3
26	0	11	3
25	1	10	2
24	0	09	8
23	0	08	7
22	0	07	9
21	1	06	6
20	0	05	10
19	0	04	20
18	2	03	44
17	3	02	124
16	3	01	1274

Figure 5. The Number of Prefix Trees

A. .COM

The domains at DL-distance 1 from COM are: BOM, CAM, CM (Cameron), CO (Columbia), MOM, and OM (Oman) The proper prefixes of COM are: CO (Colombia – available for general registration) The proper suffixes of COM are: COMCAST, COMMBANK, COMMUNITY, COMPANY, COMPARE, COMPUTER, and COMSEC

B. .EDU

.EDU does not have any proper prefix TLDs. There is one TLD at Demerau-Levenshtein distance 1: EU. This is the TLD assigned to the European Union which restricts ownership to members of the European Union. The only suffix domain for .EDU is .EDUCATION.

C. .INFO

.INFO has the proper prefix TLD .IN. .IN is the TLD assigned to India. .IN has no proper suffixes nor TLDs at Demerau-Levenshtein distance 1.

D. .NET

The domains at DL-distance 1 from NET are: BET, ET (Ethiopia), NE (Niger), NEC, NEW, NEXT, NTT, PET, and VET The proper prefixes of NET are: NE The proper suffixes of NET are: NETBANK, NETFLIX, and NETWORK

E. .ORG

.ORG has no prefix domains. There is one domain at Demerau-Levenshtein distance 1 from ORG: ONG. The only suffix domain of .ORG is .ORGANIC.

F. .RU

The domains at DL-distance 1 from RU are: AU, CU, EU, GU, HU, LU, MU, NU, PRU, RE, RO, RS, RUN, RW, SU, and VU The proper suffixes of RU are: RUHR and RUN

G. .US

Since the TLD .US is of special interest to us, we decided to include a report on it in this paper. This analysis illustrates some of the problems faced by other country specific TLDs. In particular, as noted earlier all the country code TLDs are within Demerau-Levenshtein distance 2 of each other. .US has no proper prefixes or proper suffixes. The domains at DL-distance 1 from US are: AS, BS, ES, EUS, GS, IS, LS, MS, PS, RS, UA, UBS, UG, UK, UPS, UY, UZ, and WS.

VII. The Markowsky Interceptor TLD Report

The Markowsky Interceptor TLD Report will be updated frequently and made available on my website http://DrGM.us. The goal is to keep it up-to-date so it always reflects the current state of the TLD space. The report is an alphabetical list of TLDs with a full report for each TLD. For each TLD the first line looks like: TLD number, TLD name, number of TLDS of DL-distance 1, number of proper prefixes, number of proper suffixes. The TLD number is the relative number in the space of all TLDs. The subsequent lines list the domains of Demerau-Levenshtein distance 1, the proper prefixes, and the proper suffixes. Lines with no data are omitted. Figure **??** shows some of the first domains in the report generated on July 10, 2015.

VIII. Dummy Domains

Bought 14 domain names that might be confused with Umaine.edu. The results of the confusion are summarized in Figure 7.

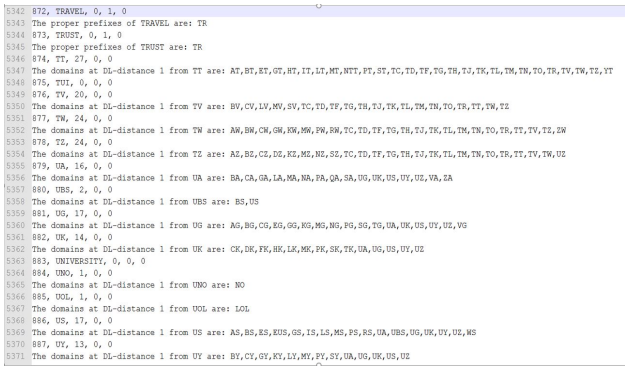

Figure 6. Sample of the Markowsky TLD Report

Dummy UMaine Domains
June 1 – September 20, 2016

Domain	Sessions	Avg. Time	Bounce	Domain	Sessions	Avg. Time	Bounce
umain.info	179	1s	99.44%	umaine.us	0		
umain.online	148	0s	94.59%	umaine.xyz	7	0s	100%
umain.org	1	0s	100%	univ...ine.info	0		
umain.us	0			univ...ine.online	8	0s	100%
umain.xyz	8	0s	100%	univ...ine.org	22	35s	86.36%
umaine.info	219	12s	91.32%	univ...ine.us	0		
umaine.online	187	<1s	90.91%	univ...ine.xyz	8	0s	100%

Figure 7. Intercepting Various UMaine Domains

IX. Metaphone Analysis

Had some correspondence with Paul Vixie and Joe St. Sauver of Farsight Security. Joe St. Sauversuggested looking at acoustic confusion. Used the Double Metaphone algorithm and came up with the following conclusions. There are 202 groups of collisions.

X. Recommendations

We consider two sets of recommendations. The first set is for people who want to reduce the chances of their communication being intercepted. The second set is for ICANN.

A. Recommendations for Defending a Domain Name

The first recommendation is that people look through the report that we make available at http://DrGM.us. It is important to examine the space of TLDs and see how the TLD you are interested in is related to other TLDs.

The second recommendation is to consider buying TLDs that might intercept your communications and direct them toward your primary TLD. Of course, if you need to buy these domains from an individual rather than from a registrar, you might find the price high. In that case you need to decide whether the potential protection is worth the price.

Third, you should consider potential TLDs from the three vantage points discussed in this paper: the prefix ordering, the Demerau-Levenshtein distance, and the possible confusion between TLDs.

Bear in mind, that even if potentially intercepting TLD-domain combinations are available, you might not be able to purchase them because they might be restricted in some way such as requiring a physical presence in a company. In such cases you might need to find an agent who can help you register these alternative domains.

B. Recommendations for ICANN

The primary recommendation here is that it is necessary to provide some separation between TLDs. It is hard to see a compelling reason for having confusing TLD pairs such as .accountant and .accountants, .ad and .ads, .auto and .autos, and .career and .careers available.

References

[1] George Markowsky, *The Problem of Interceptor Top Level Domains*, The 7^{th} IEEE International Conference on Intelligent Data Acquisition and Advanced Computiing Systems (IDAACS 2015), Warsaw, Poland, 2015.

[2] Agatha Christie, *The A. B. C. Murders*, William Morrow, ISBN-13: 978-0062073587, 2011, (originally published 1936), http://www.agathachristie.com/christies-work/stories/the-abc-murders/131.

[3] Jonathan Stempel, "UPDATE: Google Rescued Goldman Sachs In Email Debacle," *Business Insider*, July 3, 2014, http://www.businessinsider.com/goldman-suing-google-for-client-data-2014-7.

[4] Wikipedia, "Cybersquatting," March 9, 2015, http://en.wikipedia.org/wiki/Cybersquatting.

[5] Wikipedia, "Typosquatting," March 9, 2015, http://en.wikipedia.org/wiki/Typosquatting.

[6] Netcraft, "Typosquatters cashing in on .uk domains," December 2, 2014, http://news.netcraft.com/archives/2014/12/02/typosquatters-cashing-in-on-uk-domains.html.

[7] Tyler Moore and Benjamin Edelman, "Measuring the Perpetrators and Funders of Typosquatting," 14th International Conference on Financial Cryptography and Data Security, Tenerife, Spain, January 2010, http://www.benedelman.org/typosquatting/typosquatting.pdf.

[8] Garrett Gee and Peter Kim, *Doppelganger Domains*, 2011, Goday Group LLC, http://www.wired.com/images_blogs/threatlevel/2011/09/Doppelganger.Domains.pdf

[9] Robert A. Badgley, "Improving ICANN in Ten Easy Steps: Ten Suggestions for ICANN to Improve its Anti-Cybersquatting Arbitration System," *University of Illinois Journal of Law, Technology & Policy*, Vol 2001, No. 1, Spring 2001, pp. 109, http://lawlib.wlu.edu/CLJC/index.aspx?issue=yes&mainid=650&issuedate=4/1/2001&id=&abstracts=yes&coins=yes&homepage=yes&jf1=no&jf2=no.

[10] J. Ryan Gilfoil, "A Judicial Safe Harbor under the Anti-Cybersquating Consumer Protection Act," *Berkeley Technology Law Journal*, Vol. 20, No. 1, Article 22, pp. 185-208. http://scholarship.law.berkeley.edu/cgi/viewcontent.cgi?article=1511&context=btlj.

[11] Wei-erh Chen, "Optimizing Online Trademark Protections Given the Proliferation of Generic Top Level Domains," *Iowa J. Corp. L.*, Vol. 38, Spring 2013, p. 585.

[12] Wikipedia, "Damerau–Levenshtein distance," March 21, 2015 http://en.wikipedia.org/wiki/Damerau\%E2\%80\%93Levenshtein_distance.

[13] Navarro, Gonzalo, "A guided tour to approximate string matching," ACM Computing Surveys 33 (1): 31–88, 2001, doi:10.1145/375360.375365, http://google.com. "How misaddressed emails can make you vulnerable,"