Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Telecommunications
Laboratory of Cryptography and System Security (CrySyS Lab.)

Szurdi János

UNDERSTANDING THE PURPOSE OF DOMAIN REGISTRATIONS

SUPERVISOR

Dr. Félegyházi Márk

CrySyS Lab.

BUDAPEST, 2012

Table of contents

Table of contents **2**
Összefoglaló **6**
Abstract **7**
1 Introduction **8**
2 Background **11**
2.1 DNS 11
2.1.1 Brief History of DNS 11
2.1.2 DNS Structure 12
2.1.3 Domain Names 15
2.2 Usage of domain names 17
2.2.1 Speculative registration of domain names 17
2.2.1.1 Cybersquatting 17
2.2.1.2 Typosquatting 18
2.2.1.3 Domain name front running 19
2.2.1.4 Domain tasting 19
2.2.1.5 Domain parking 19
2.2.1.6 Defensive registrations 20
2.2.2 Malicious registration of domain names 20
2.2.2.1 Spam 20
2.2.2.2 Phishing 21
2.2.2.3 Spamming and Malware distribution through botnets 22
3 Related Work **23**
3.1 Understanding domain registration abuses 23
3.2 Measuring the perpetrators and funders of typosquatting 23
3.3 PhishDef: URL Names Say It All 24
3.4 EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis 25
3.5 Correlating Domain Registrations and DNS First Activity in General and for Malware 26
3.6 Mining DNS for Malicious Domain Registrations 27
4 Methodology and framework **29**
4.1 Modular system design 29

4.2 Information sources on domain names 31

4.2.1 The domain name itself – lexical analysis 31

4.2.2 Zone file 32

4.2.3 Whois information 33

4.2.4 Content analysis 35

4.2.5 Passive DNS data 35

4.2.6 Online directories 35

4.2.7 Blacklists 36

4.2.8 Comparison of the information sources 36

4.3 Domain sets used for analysis 38

4.3.1 Regular domain sets 38

4.3.2 Malicious domain sets 38

4.3.3 Benign domain sets 38

4.4 Heuristical modules and conclusions 40

4.5 OpenDNS for direct categorization 41

5 Usage of com domains 44

5.1 Wiki Module 44

5.1.1 Design of the Wiki Module 44

5.1.2 Random Wiki Crawler 46

5.1.3 Bigram Module 47

5.1.4 Best Overlap Module 48

5.1.5 Wiki Tagger Module 48

5.1.6 Analysis 49

5.2 Random Seeker Module 58

5.2.1 Design of the Random Seeker Module 58

5.2.2 Evaluation of Results 59

5.3 Conclusions and future improvements 62

6 The BIZ Top-Level Domain: Ten Years Later 64

6.1 Motivation 64

6.2 Data and Methodology 65

6.3 Analysis 67

6.4 Discussion and conclusions 70

7 Summary 72

Acknowledgements 74

Bibliography 75

HALLGATÓI NYILATKOZAT

Alulírott **Szurdi János**, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2012. 01. 11.

....

Szurdi János

Összefoglaló

Az Internet infrastruktúrájának egyik alapvető pillére a Domain Name System (DNS) és a hozzá kapcsolódó protokollok. Minden hálózati eszközt az IP (Internet Protokoll) címe alapján lehet elérni az Interneten, de ez a cím az emberi szem számára nehezen értelmezhető. A domain nevek bevezetésének célja az volt, hogy a felhasználók számára megkönnyítse az Internetes erőforrások elérését. Napjainkra kialakult egy profitorientált, spekuláns piac a vonzó domain nevek felvásárlására vagy a lejáró, könnyen megjegyezhető domain nevek megszerzésére. A regisztráció könnyedsége vonzóvá teszi rosszindulatú szereplők számára, hogy visszaéljenek a rendszerrel. Mivel a domain regisztrálás ára nagyon alacsony, és a domain registrar-ok működése alig van szabályozva, így a rosszindulatú szereplőknek lehetősége van a domainek tömeges regisztrálására. Feltételezések szerint a domain regisztrációk nagy része rosszindulatú céllal történik, de ez nem bizonyított.

Internet Corporation for Assigned Names and Numbers (ICANN) nevű irányító szervezet 10 évvel ezelött vezette be a com Top Level Domain (TLD) mellé a biz és info gTLD-ket (generic Top Level Domain) azzal a céllal, hogy a domain név regisztrálók választási lehetőségét bővítse, illetve, hogy az akkori domain név struktúrából kizárt entitásoknak lehetőséget adjon a belépésre. 2011-ben, 10 évvel később joggal tesszük fel a kérdést, hogy valóban sikerült-e a kitűzött célt elérni az új gTLD-kel vagy az újítás csak a már létező márkatulajdonosokat kényszeríti defenzív regisztrációra és a spekuláns szereplőknek ad teret, hogy a domain név regisztrációkból szerzet profitjukat növeljék?

A kutatás célja, hogy egy keretrendszerben összesítsük a néhány már létező domain regisztrációval foglalkozó tanulmányt, és hogy ezt a rendszert kibővítsük, finomítsuk. Minket az eddig létező kutatásokkal szemben a domain nevek általános kategorizálása érdekel, különös tekintettel a domain nevekkel való visszaélésekre. A jelenlegi domain regisztrációs célok lényegét szeretnénk megérteni. A metodológiát és tervezési megoldásokat úgy fogjuk felépíteni, hogy a domain neveket több szempont alapján is kategorizálhassuk majd. Cél a fentebb feltett kérdések megválaszolása, a domain nevek regisztrációjának céljával és az új gTLD-k bevezetésével kapcsolatban.

Abstract

The Domain Name System (DNS) and its protocols are one of the fundamental building blocks of the Internet infrastructure. Each machine on the Internet is reached via its Internet Protocol (IP) address, but this numerical address is unreadable for human users. Domain names were invented to help a human-readable identification of Internet resources. However, nowadays there is a substantial speculative market to register appealing new domain names or to obtain easy-to-remember domain names that expire. The ease of registration is also inviting miscreants to abuse the system. Because the price of registering a domain is extremely low and domain registrars loosely regulated, miscreants can register domain names in bulk. Experts suspect that most of the new domain names are malicious, but no evidence supports this claim.

It has been 10 years since Internet Corporation for Assigned Names and Numbers (ICANN) announced the introduction of new generic Top Level Domains (gTLD) biz and info besides the already existing com TLD with the intention to broaden consumer choice and make opportunities for entities that have been shut out under the name structure existing at that time. In 2011, ten years later, the question is if biz gTLD has evolved into the role intended or whether it has merely induced defensive registrations by existing trademark holders who already had equivalent com domains.

Our research aims at summarizing the existing literature in a general framework and also extending these papers beyond their scope. We are interested in the generic categorization of domain registrations not neglecting the potential for abuse. We aim at understanding the essence of current domain registration practices. We will develop the methodology and design solutions to categorize domain name registrations for various purposes. The goal is to answer the above questions about the introduction of new gTLDs and the purpose of domain name registrations.

1 Introduction

Each machine or interface can be reached by another machine or interface on the Internet via its IP (Internet Protocol) address. However, IPv4 numerical addresses are hard to read and hard to remember for humans and the problem becomes worse with IPv6 addresses. Domain names were created to provide a meaningful translation of IP addresses for the human users of the Internet applications and to enable the change of IP addresses without changing the reference point to the machine. Domain names constitute the basis of web browsing and email, the most important applications on the Internet. This makes it possible to change the location of the Internet resource without changing the domain name.

In 1983, when Jon Postel and Paul Mockapetris invented the DNS (Rader, 2001), probably didn’t think of how much the purpose of domain name registration will change from the original intent. Nowadays even benign actors carefully considers which domain to choose from business aspect, because it is very important to choose an easy-to-memorize domain name that also sounds good to help popularizing the product or service, the main purpose of the registration. Not all domain registrations happen with good intention to make a name or brand available via the Internet. Speculative registrations typically happen with the intention of selling domain names for profit or monetizing collateral web traffic of popular websites. Parked domains can be used to make profit from accidental traffic using advertising. In a more questionable form of typosquatting or cybersquatting traffic is often redirected to concurrent brands’ pages. The ease of registration is also inviting miscreants to abuse the system. Because the price of registering a domain is extremely low and domain registrars loosely regulated, miscreants can register domain names in bulk. Spammers need large number of domains to avoid domain blacklisting, when a set of domain names get blacklisted, they simply jump to the next set and use this new set for their malicious campaigns. Domains are used for phishing as well. In phishing, legitimately looking websites are used to lure victims into giving out their personal data to the cyber criminals. Experts suspect that most of the newly registered domain names are malicious, but no evidence supports this claim. From com domains, ten thousands of domains are registered and deleted daily. In this thesis, we want to distinguish active domains from inactive and malicious domains from benign ones.

In 2001, ICANN (Internet Corporation for Assigned Names and Numbers) announced the introduction of new gTLDs (generic Top Level Domain) biz and info with the intention to broaden consumer choice and to open door for entities that did not have the chance to secure their brand domain name for com. The biz gTLD, in particular, has been intended as an alternative option to the popular com top-level domain. In 2011 ten years later the question is if biz gTLD has evolved into the role envisioned by ICANN and became a viable alternative to com giving trademark holders, who are unable to register a com name, an attractive substitute or whether it has merely induced defensive registrations by existing trademark holders who already had equivalent com domains? The question has become even more actual now, when ICANN is being sued for the new gTLD xxx and when ICANN plans to introduce a more open system where anybody can have their own gTLD if willing to pay the substantial entry price. We want to assess the question of biz to perceive if introducing new gTLDs going to increase consumers' choice or just force trademark holders to register defensively and give more opportunities to speculators and criminals.

In Chapter 2, I will give an introduction on the DNS system. Brief history of DNS will tell us why it was important to start using domain names and to make a global directory of domain names called Domain Name System. After a summary on DNS history, Section 2.1.2 will give us an insight to how DNS works. The introduction to DNS is completed by an overview on domain names. In the following section, I will provide a detailed discussion of domain speculation methods and malicious domain registrations. I will present examples for both malicious and speculative registrations, with some real life cases.

In Chapter 3, I will discuss related work focusing on the sources, methodologies and results most fitting to benefit us in our case studies. For each case study, the motivation will be considered first, then the source they used, and how did they use it. Finally, the results will be deliberated.

Chapter 4 introduces a framework designed to be flexible, modular and robust in a way of letting mass data analysis to be performed. First the problem of categorization will be discussed more thoroughly, then the system design will be introduced with the intended purpose, and then the methodology used for the research.

Also in Chapter 4, I will discuss my data sources that included benign, malicious, and average domain names used for categorizing.

Chapter 5 presents the study on the usage of com domains. Here we shall go through the design and evaluation phases, to show you the tools we used. Chapter 6 will give some more details on ICANN and gTLDs, showing you the analysis and consequent results. Finally, in Chapter 7, I will summarize my results and present potential improvements on how the results could be improved. I will also sketch the road towards an integrated toolbox to fully categorize new domain registrations.

2 Background

2.1 DNS

2.1.1 Brief History of DNS

Next, we present a concise summary of the history of DNS. This summary is largely based on “One history of DNS” by Ross Rader (Rader, 2001), and I also used Wikipedia article on DNS (Wikipedia: Domain Name System, 2011).

The Domain Name System (DNS) is one of the fundamental protocols of the Internet infrastructure. As it provides a fundamental service, it is important to understand how it evolved from its inception until today. History of DNS will be observed focusing on the significance of giving host names to machines and making a global directory structure, focusing on the expansion of the domain name hierarchy.

In 1967, when Doug Engelbart created the ARPANET NIC (Network Information Center), DNS did not exist yet. At that time the network was so small that users and servers knew how to get around from service to service and interact with each other, so a global directory service was not needed.

As the network grew in size it became clear that such a service would be important. Nevertheless, it did not happen yet, in 1971 Peggy Karp conceived “host mnemonics”, or simply, Internet names. In RFC 226, she created the concept of a lookup table that mapped all of the network resources. It was called HOSTS.TXT and it contained all hostnames and their IP addresses. Whenever a new machine was added to the network the operator had to send the appropriate information to SRI (Stanford Research Institute), where it was added to the next release of the HOSTS.TXT, which was stored on a global ftp server.

However, the network expanded further and the size of HOSTS.TXT grew in a direct relationship creating a scalability problem, when HOSTS.TXT got too big. There was a problem too when operators did not update their HOSTS.TXT files on a regular basis, and this led to name collisions. The rapid growth of the network made a centrally maintained, handcrafted HOSTS.TXT file unsustainable; it became necessary to implement a more scalable system capable of automatically disseminating the requisite information. At the request of Jon Postel, in 1983 Paul Mockapetris invented the DNS and wrote the first implementation. RFC 920 is important for us; it has outlined the initial top-level domain names that would be added to the DNS when it was finally deployed. This included com, net, org, edu, gov, mil and arpa. The document has also outlines plans for the creation of country- related TLDs using the two letter ISO (International Organization for Standardization) codes (becoming, ca, us, uk, hu, etc.). Finally, in March of 1985 the first domain names were registered.

In November 1988, another TLD was introduced, int. This TLD was introduced in response to NATO's request for a domain name that adequately reflected its character as an international organization.

By the time ARPANET retired in 1990, the network of networks had grown to include over 100,000 connected host computers. Until 1995, academic policy on the name-space allowed anyone having access to a name server to register a domain name with the NSI InterNIC – for free. Needless to say, as the perceived value of being online increased, so did the perceived value of Internet Domain Names. Thus, warehousing and speculation was born. It was not unusual for a speculator to register hundreds, or even thousands of domain names simply based on the potential that someone might want to purchase the domain name from them in the future. After all, the names were free to the first-comer to register, why not grab as many as you could on the off chance you could make a few bucks. Time passed by and registering domains did not stay free, but until now speculative domain registration remained a profitable endeavor.

In 2001 ICANN (Internet Corporation for Assigned Names and Numbers) announced the introduction of new gTLDs (generic Top Level Domain) aero, biz, coop, info, museum, name, and pro with the intention to broaden consumer choice and make opportunities for entities that have been shut out under the name structure existing at the time.

In 2011, ten years later, ICANN introduced a new top-level domain xxx, and decided on introducing a new open architecture of domain registrations. In the new system anyone can setup a registry for a new TLD for a price of 185 000 dollars plus technical setup for 100 000 dollars and upkeep could be another 100 000 dollars per year (Pepitone, 2011).

2.1.2 DNS Structure

Writing about the DNS Structure I relied on Computer Networking by Kurose and Ross (Kurose & Ross, 2009).

DNS uses a large number of servers, organized in a **hierarchical** fashion and distributed around the world, in order to be scalable and robust. This means there is not a single DNS server that would know all the mappings for all of the hosts in the Internet. We can identify three classes of DNS servers: **root DNS servers**, **TLD DNS servers** and **authoritative DNS servers**. They are organized in a hierarchy, shown in Figure 2-1.

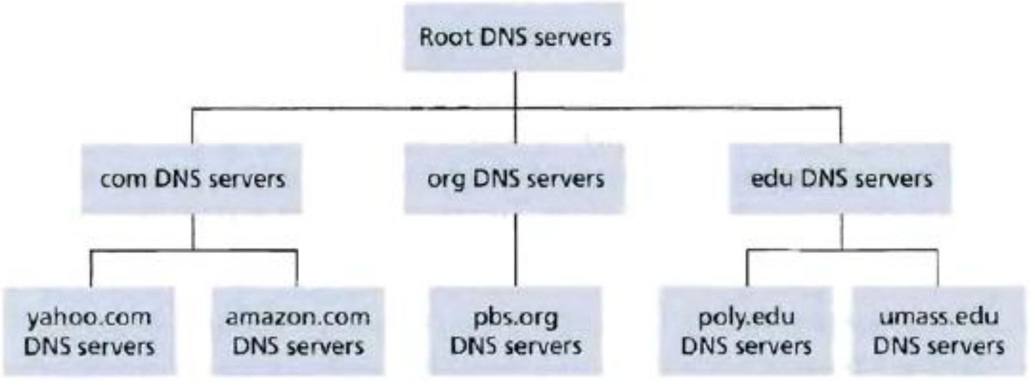


Figure 2-1, hierarchy of DNS servers (Kurose & Ross, 2009)

Root DNS servers are used for finding authoritative name servers responsible for TLD zones. There are 13 Root DNS servers on the Internet (labeled A through M). It does not mean there are only 13 servers physically; each operator uses redundant computer equipment to provide reliable service even if failure of hardware or software occurs. Nine of these servers operate on several different geographical locations, using anycast addressing (Wikipedia: Root name server, 2011).

TLD DNS servers are responsible for both generic top-level domains such as com, net, biz; and country-code top-level domains such as hu, uk, de, ru, and so on. Everyone with publicly accessible hosts (such as Web servers and mail servers) on the Internet must provide publicly accessible DNS records that map the names of those hosts to IP addresses.

Authoritative DNS servers are housing these DNS records. Every DNS zone must be assigned a set of authoritative name servers that are installed in NS records in the parent zone.

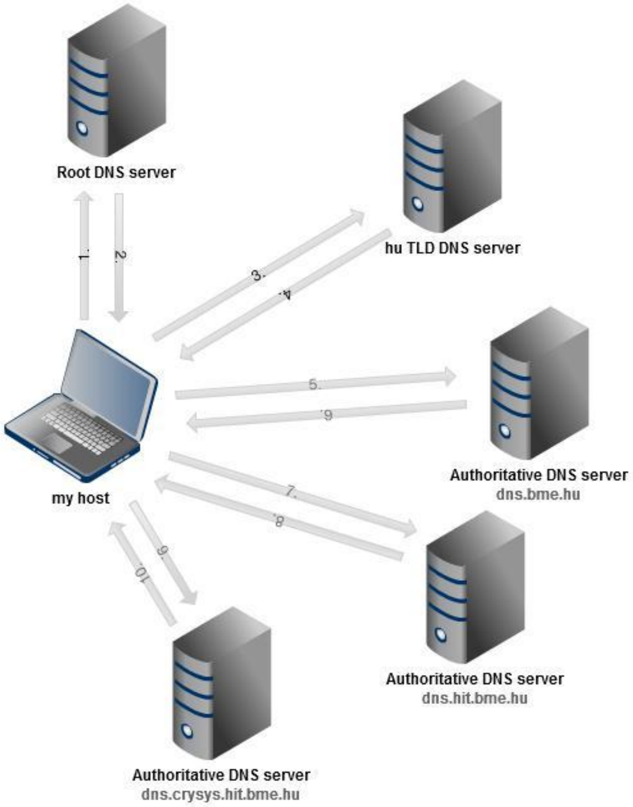


Figure 2-2 DNS query example

For a quick example, let’s see what would happen if our host would like to get the IP address of crysys.hit.bme.hu. In our case, we suppose that bme.hu has an authoritative name server called dns.bme.hu, hit.bme.hu has an authoritative name server called dns.hit.bme.hu and for crysys.hit.bme.hu, there is another authoritative name server called dns.crysys.hit.bme.hu. Every host is configured to know the IP addresses of the root DNS servers (or at least to know the IP address of a Local DNS server, which knows the addresses of the root DNS servers). As shown in Figure 2-2 our host first sends a DNS query message to a root DNS server. The root DNS server detecting the hu suffix in the query, returns a list of IP addresses of TLD servers responsible for hu TLD. The host will send a DNS query to one of these TLD servers, which will answer with the IP address of dns.bme.hu. In the next step our host queries dns.bme.hu, which will answer with the IP address of dns.hit.bme.hu. Then we get the IP address of dns.crysys.hit.bme.hu in the same way, and our last DNS query will go to dns.crysys.hit.bme.hu which will sends us the IP address of crysys.hit.bme.hu.

2.1.3 Domain Names

Domain names are the human-memorizable representation of Internet resources. This abstraction makes possible to move any resource to a different physical location in the address topology of the network. Generally, a domain name represents an IP resource, such as a server hosting a website, or the website itself. Domain names also used to identify ownership or control of a resource, like in SIP (Session Initiation Protocol) or in emailing (Wikipedia: Domain name, 2011).

The first commercial domain names were registered on the March of 1985, some claim the very first com domain was symbolic.com, and others claim think.com (Rader, 2001). By 1992, fewer than 15,000 *com* domains had been registered. In December 2009, there were 192 million domain names (Wikipedia: Domain name, 2011). The largest fraction of them is the com domain, nowadays containing nearly 98 million domain names (HosterStats.com).

Domain names consist of two or more parts delimited by dots, see our example: crysys.hit.bme.hu. The rightmost part is called the **top-level domain** name in our example the hu is the top-level domain. The hierarchy of domains descends from the right to the left label in the name; each label to the left specifies a subdivision, or **subdomain** of the domain to the right. A **hostname** is a domain name that has at least one associated IP address. In our example, crysys.hit.bme.hu is a host name while hu is not if there is no IP address associated to it. Below the top-level domains in the domain name hierarchy are the **second-level domain** (SLD) names, which is bme in crysys.hit.bme.hu.

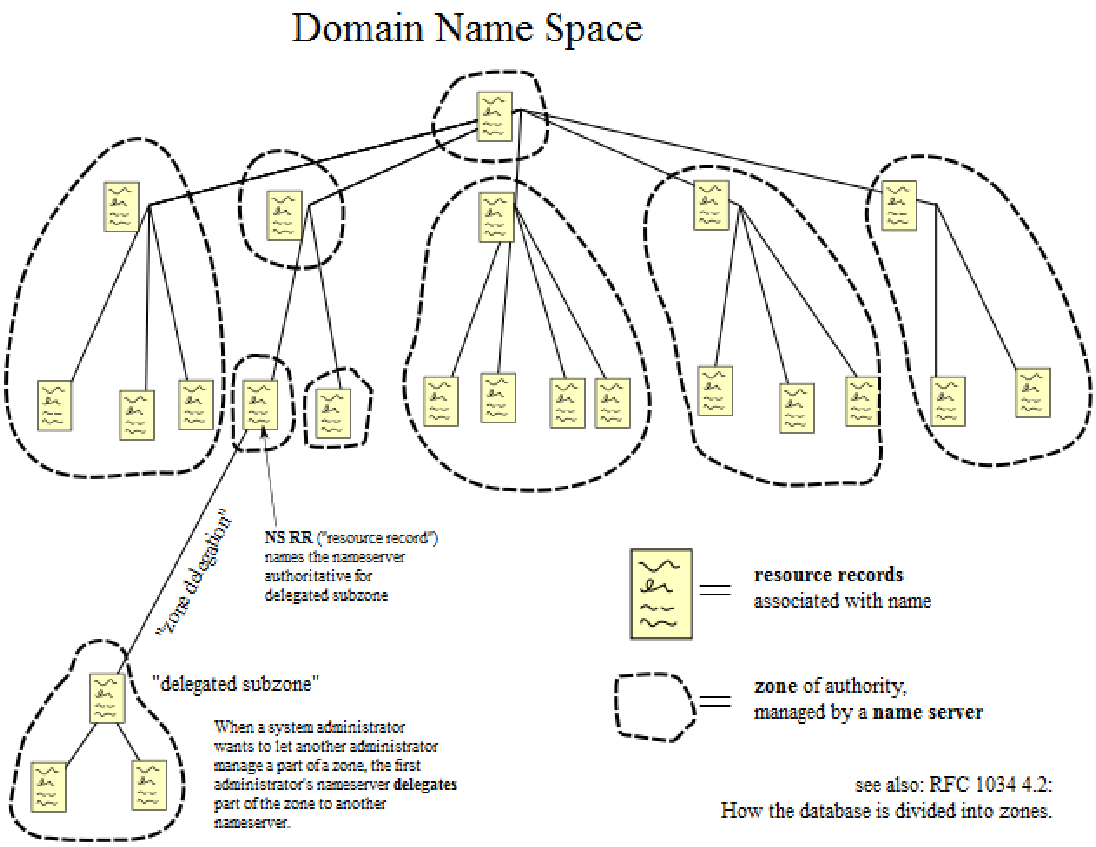

Domain Name Space
"zone delegation"
NS RR ("resource record") names the nameserver authoritative for delegated subzone
"delegated subzone"
When a system administrator wants to let another administrator manage a part of a zone, the first administrator's nameserver delegates part of the zone to another nameserver.
= resource records associated with name
= zone of authority, managed by a name server
see also: RFC 1034 4.2: How the database is divided into zones.

Figure 2-3 Domain Name Space (Wikipedia: Domain name, 2011)

There are two kind of TLDs **generic TLDs** (gTLD) and **country-code TLDs** (ccTLD). The gTLDs are: com, net, org, edu, gov, mil, arpa, int, aero, biz, coop, info, museum, name, pro and from now on xxx. As mentioned earlier the ccTLDs are using the two letter ISO codes of the countries as hu, de, us, uk and so on.

Interesting data on domain names are that from being free to register a domain, the value of some domains has rocketed quite high. Some of the most expensive domain sales on record (Wikipedia: Domain name, 2011) are:

1. Insure.com 2009 $16 million
2. Sex.com $14 million in October 2010
3. Fund.com 2008 £9.99 million
4. Porn.com 2007 $9.5 million
5. Fb.com $8.5 million in November 2010
6. Business.com $7.5 million in December 1999
7. Diamond.com 2006 $7.5 million
8. Beer.com 2004 $7 million

9. Israel.com 2008 $5.88 million
10. Casino.com 2003 $5.5 million

2.2 Usage of domain names

The categories of speculative and malicious domain registrations are not disjoint sets. For example, cybersquatting and typosquatting are illegal by law. According to the United States federal law known as the Anticybersquatting Consumer Protection Act, cybersquatting is registering, trafficking in, or using a domain name with bad faith intent to profit from the goodwill of a trademark belonging to someone else.

Domain speculation is the registration of a domain name with the intention of reselling it for a profit at a later date or generating ad revenue from type-in navigation traffic. In Section 2.2.1 we discuss the registration/usage types that satisfy the definition of domain speculation, with a particular attention to malicious domain registrations. If the malicious registration/usage is not speculative by the above definition above, it will be discussed in Section 2.2.2. Personal note: when speculative domain registrations receive unintended traffic originating from type-in navigation, it is always a traffic taken from the rightful target, making these kinds of registrations malicious.

2.2.1 Speculative registration of domain names

2.2.1.1 Cybersquatting

The practice of registrants making profit from registering domain names matching the name of another company, product or trademark is called cybersquatting. The term is derived from "squatting", which is the act of occupying an abandoned or unoccupied space or building that the squatter does not own, rent or otherwise have permission to use (Wikipedia: Cybersquatting, 2011).

Moore and Edelman discussed in their paper on typosquatting (Moore & Edelman, 2010) many strategies that cybersquatters use to profit from their registrations. After grabbing particularly valuable domains, some squatters sought small ransom from the organizations desiring those domains. Coercing original trademark holders to pay the ransom, the cybersquatters would redirect traffic to competition or to compromising pages, making the blackmailed organization look bad. They would even put damaging content on the squatted domain. In one notorious case, a squatter redirected thousands of expired domains to adult websites, making it all the less palatable to leave the domains with the squatter, and all the more tempting to pay to get the domains back. Other squatters redirect the domain to the competition of the cybersquatted victim. In these cases, the squatter could profit directly if he operated the landing page, or indirectly through marketing commissions paid by the destination site. Finally, a growing share of squatters found profits through advertising, typically, showing pay-per-click ads through the web's top ad networks using visitors of the original trademark as the target group.

A good example of cybersquatting was when Heelys shoes were already popular in the U.S.A. and it was about to enter the untouched market in Hungary. Two local firms were competing for the exclusive right to distribute it in Hungary. One of them registered the heelys.hu domain, but the other one obtained the right for distributing Heelys shoes in Hungary, exclusively. The owner of the heelys.hu domain did not give it up, but asked a high price for the domain. The other one did not want to pay. After a while, the domain owner started distributing replicas of the original Heelys shoes and redirected heelys.hu domain visitors to his own, replica shoes selling, site.

2.2.1.2 Typosquatting

The article of Moore and Edelman on Typosquatting was used for this part (Moore & Edelman, 2010). Typosquatting is a special form of cybersquatting when squatters intentionally register misspelled name variations of popular websites anticipating that users will often mistype those domains and will reach the squatters' sites. Besides typing errors and common misspelling, differently phased domain names or using different TLD can also be tools of typosquatters to divert traffic from the original site. Monetizing typosquatting can be done many cunning ways, just as the more general cybersquatting. The most popular techniques are: advertising with pay-per click ads, linking or redirecting to competition, and defensive registrations.

We can see numerous examples of typosquatting some discussed on Wikipedia (Wikipedia: Typosquatting, 2011). Many companies have reputation for ruthlessly chasing down typosquatted names, including Verizon, Lufthansa, and Lego. Lego, for example, has spent roughly $500,000 USD on taking 309 cases to UDRP (Uniform Domain-Name Dispute-Resolution Policy). Celebrities, from singers to star athletes have also frequently battled and protected their domain names. Prominent examples include basketball player Dirk Nowitzki's UDRP of DirkSwish.com and actress Eva

Longoria's UDRP of EvaLongoria.org. A complainant in a UDRP proceeding must establish three elements to succeed:

- The domain name is identical or confusingly similar to a trademark or service mark in which the complainant has rights;
- the registrant does not have any rights or legitimate interests in the domain name; and
- the registrant has registered the domain name and is using it in "bad faith."

2.2.1.3 Domain name front running

Domain name front running as described by Coull, White, Yen, Monrose, & Reiter (Coull, White, Yen, Monrose, & Reiter, 2010) is the practice whereby a domain name registrar uses insider information to register domains for the purpose of re-selling them or earning revenue via ads placed on the domain's landing page. By registering the domains, the registrar locks out other potential registrars from selling the domain to a customer. The registrar typically takes advantage of the 5-day "domain tasting" trial period, where the domain can be locked without payment. In January 2008 it was reported that Network Solutions uses data collected from their web-based WHOIS search to register every domain that users check for availability (Wikipedia: Domain name front running, 2011).

2.2.1.4 Domain tasting

Based on the article Understanding Domain Registration Abuses (Coull, White, Yen, Monrose, & Reiter, 2010), domain tasting means that a registrar is allowed to delete a domain within five days of the initial registration at no cost, and it is also known as the add grace period. This policy can be easily abused by registrars and registrants alike in order to gain information about the value of a domain via traffic statistics taken during the grace period.

2.2.1.5 Domain parking

A parked domain is one which is not in active use by the registrant, and which does not represent a name or brand used by the registrant. Parked domains are typically held with the intention of selling them at a profit, and monetizing accidental Web traffic with advertising (Halvorson, et al., 2012).

One way to monetize web-traffic is to have pay-per click ads on the website. Pay-per click ad is a form of advertisement where the advertiser only pays when someone clicks on the advertisement and goes to his page. This is a good business model for both advertisers and domain parkers, because advertiser pays only for the traffic really hitting his page, and domain parkers gets money from traffic they did not initiate with investment. Usually links, shown on a parked domain, change based on how much hits they get from the actual domain.

Domain names have to be reregistered from time to time. If a domain is not reregistered in due time, it can be registered by anyone else. Registrars can get hold of expired domains, and gain huge traffic, because traffic going towards a domain will not stop immediately. Inbound links will point to the domain until website operators and search engines start to remove these links (Wikipedia: Domain parking, 2011).

2.2.1.6 Defensive registrations

In the paper, we wrote on biz TLD (Halvorson, et al., 2012) we described defensive registration. The purpose of a defensive registration is to prevent another party from either misrepresenting itself as the registrant or from simply capturing traffic (intended for the registrant) for advertising purposes. A defensively registered domain is not used, either internally, or externally to identify products, services, or network infrastructure. The difference between a defensive registration and cybersquatting or typosquatting (registering misspellings of popular brands): if, the registrant is also the owner of the brand name or trademark, the registration is defensive; if the registrant is a third party with no legitimate claim to the name, the registration is cybersquatting or typosquatting.

2.2.2 Malicious registration of domain names

2.2.2.1 Spam

By spam, we do not mean the canned precooked pork meat, but the use of electronic messaging systems to send unsolicited bulk messages indiscriminately. Spamming stays economically viable because of the low operational cost and because it is difficult to hold spammers accountable for mass mailing (Wikipedia: Spam (electronic), 2011).

Spamming is a serious problem: according to assessments, 88-90% of all email is spam (MAAWG: Email Metrics Program, 2011). The price of sending out trillions of e-mails needs an infrastructure that is maintained by the internet service providers; but eventually the end users pay the price of spam.

Spamming is used for various purposes: mass, unsolicited advertisement, phishing, and malware distribution. Independently of the purpose, spammers usually put a link in the email for the recipient to click and to be directed to a webpage. This is the point where domains play a crucial role. Due to the improvement in spam defense, notably blacklisting of resources, spammers need to register many domains to maintain their operation.

2.2.2.2 Phishing

Phishing targets the weakest chain in financial transactions, the human users. Security is not user proof; security systems assume appropriate behavior of the user. The intention of phishing is to steal users' personal data such as username, e-bank password, credit card number, pin code etc. This kind of data has a great value on the black market, and phishers usually pass it on to the so-called "cashiers", who will make money from the stolen data.

One way of phishing is, when miscreants maintain a webpage mimicking a real bank or e-bank site; this activity is called website forgery. Spammers direct people to the forged website by sending out spam e-mails to the target group. Customers can easily be fooled to believe that the e-mail is coming from the bank. Composing and formatting the message to look official, spoofing senders e-mail address (changing it to it@mybank.com), and manipulating the link to the forged website so it looks like a link to the original website (<a href="http://phisherssite.com"> http://www.ithelp.mybank.com<a/>).

Good example by Wikipedia (Wikipedia: Phishing, 2011) of tricking users into giving out their personal data is an experiment made in June 2004, 500 cadets of West Point Academy was sent fake e-mails, and 80% of the students gave out some personal information. Social engineering techniques such as phishing could be quite dangerous and can only be avoided only by educating and training the human participants.

Phishers need to register many domains not just for the spamming they make, but also for the forged websites. The purpose of mass registering domain names is same as in spamming, to avoid blacklisting.

2.2.2.3 Spamming and Malware distribution through botnets

Cybercriminals have software and infrastructure to control millions of machines on the Internet. Computers infected by these kinds of viruses are called zombies, the networks of zombie machines are called botnets, and the up-keepers of these botnets are called the herders. Part of botnets, even millions of zombie machines can be hired for spamming, spreading malware or making Distributed Denial of Service (DDoS) attacks. As Figure 2-4 shows five botnets are responsible for 74% of spam.

Botnets have a high value and herders make various steps to protect their herd of zombies. One such technique is, when the IP address related to a domain is constantly changing; it is called IP-fluxing or fast-fluxing. With this technique, hundreds or thousands of IP addresses are associated with a domain name. Domain flux is effectively the inverse of IP flux and refers to constant changing and allocation of multiple FQDN’s to a single IP address (Ollmann, 2009). Subsequently botnet owners need a great number of domain names to avoid discovery and counter-measurements.

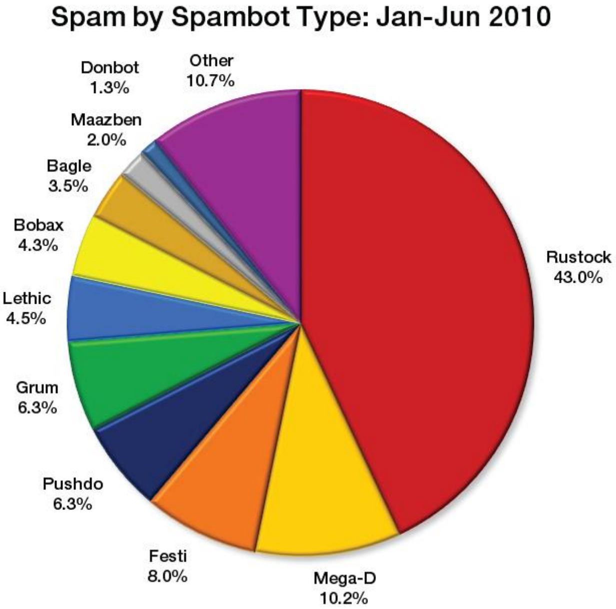


Figure 2-4 Botnets responsible for spam (M86security.com: Security Labs Report, January -June 2010 Recap, 2010).

3 Related Work

3.1 Understanding domain registration abuses

First related work we discuss is the paper called “Understanding domain registration abuses” (Coull, White, Yen, Monrose, & Reiter, 2010).

Many users assume that typing intuitive keywords based domain names into the address bar of the browser will direct them to the desired page. This habit (called type-in navigation) made many domain names quite valuable. It has bred domain speculations described in section 2.2.1. Though speculation is technically allowed by ICANN rules, it has led to many abusive behaviors. This paper reviews domain name speculation, domain tasting and domain front running.

Data collected by Google via its Insights for Search and Trends services is used to find hot topics. These services rate and rank the top searches made by users over a given time frame, and provide up to ten related searches for each. The paper writers assume the searches users make on Google about an event or topic are closely related to the domain names they would navigate to using type-in navigation. In this paper, the authors use this data to make regular expressions for finding relationships between new domain registrations and hot topics (established by Google service).

They found 15954 domains in 113 topics verified to be directly related to the topic at hand. Their research has also shown that speculators clearly prefer some registrars to others. They determined that 76% of distinct new registrations are the result of domain tasting, 66% of speculated domains were registered only for the domain tasting five day grace period. For domain front running, their analysis shows that none of the observed registrars is associated with a statistically significant increase in the registration rate of queried domains.

3.2 Measuring the perpetrators and funders of typosquatting

Moore and Edelman wrote a very interesting paper on Typosquatting (Moore & Edelman, 2010) which we discuss in this section.

The aim of this paper is to find typosquatting domains targeting popular website addresses and find the way, how squatters gain revenue from these domains. Typosquatting means intentionally registering misspelled variations of popular websites in anticipation that users mistype those domains and reach squatters’ sites (more details in Section 2.2.1.2).

Damerau-Levenshtein distance: the minimum number of insertions, deletions, substitutions or transpositions required to transform one string into another. Fat-finger distance is: the minimum number of insertions, deletions, substitutions or transpositions using letters adjacent on a QWERTY keyboard to transform one string into another.

The popular domains are chosen from Alexa’s ranking. From the most popular 6000 domains, 3264 are at least five characters long com domains and these were used for the study. In the next step, typosquatting domains were generated from the chosen 3264 domains using Damerau-Levenshtein and fat-finger distance up to two. They found 1.910.738 candidate typos from the 81 million registered com domains at that time. With false positive estimates from manual checks, they appraised the number of com typo domains targeting the popular sites equals approximately 938 000 domains.

After crawling more then 250 000 of these 938 000 typo domains, they found that 80% are supported by pay-per click ads often advertising the correctly spelled domain and its competition. Another 20% include static redirection to other sites. They found that typosquatting is highly concentrated: Of typo domains showing Google ads, 63% use one of five advertising IDs, and some large name servers host typosquatting domains as much as four times as often as the web as a whole.

“We suspect typosquatting will continue so long as advertisers and ad networks continue to fuel and fund these practices. But let no one suggest identifying typo domains is impossible: The overwhelming majority of typos are easy to recognize, by hand or using straightforward automation. At the same time, with typo domains highly concentrated at a few large domainers and ad platforms, intermediaries could significantly discourage the registration and use of typo domains if they were so inclined.”

3.3 PhishDef: URL Names Say It All

Faloutsos, Markopoulou, & Le introduce a system in their paper (Faloutsos, Markopoulou, & Le, 2011) on finding URLs used for phishing

Phishing is, when cyber criminals try to steal personal user information by mimicking websites or sending fake e-mails. For more details, see Section 2.2.2.2. In their paper, the authors construct a Phishing URL detector using lexical properties.

They use many sources such as PhishTank, MalwarePatrol, Yahoo Directory, and DMOZ for creating malicious and legitimate domain sets. During the study, five kinds of features were used:

- features related to the full URL,
- the domain name,
- the directory,
- the file name,
- the argument part.

These features were used to detect common obfuscation techniques worked by attackers: obfuscating the host with an IP address, obfuscating host with another domain, obfuscating host with large host names, and domain unknown or misspelled.

This set of experiments show that using lexical features alone leads to comparable classification accuracy to using full features with only a 1% difference. The high accuracy and the lightweight properties of lexical features make a strong case for using them alone.

3.4 EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis

In this section we discuss a novel work (Bilge, Kirda, Kruegel, & Balduzzi, 2011) aiming to find malicous domains using passive DNS analysis.

The domain name system is abused many ways to support malicious activities; some of these are described in Section 2.2.2. The paper discussed in Section 3.4 aims to use passive DNS data to find such malicious activities. They bring up two examples of activities, which could be found using their technique; one is botnets resolving domain names to locate their command and control center, the other example is spam mails containing URLs that link to domains that resolve to scam servers.

Passive DNS analysis means gathering DNS traffic from DNS servers. Allowing the classifier to work well, large amount of training data was collected. Then offline analysis was performed on this data and was used to determine DNS features that can be used to distinguish malicious DNS qualities from benign ones. Benign domain set was gathered using Alexa and Whois, malicious data was from blacklists such as malciousdomains.com or PhishTank. Architecture of EXPOSURE is built up from five components: Data Collector component, Feature Attribution component, Malicious and Benign Domains Collector, Learning Module component, and the Classifier component.

For the analysis, interesting time based features were used. They tried to detect abrupt changes; their insight was that malicious domains often show a sudden increase in the number of requests, and then the requests suddenly disappear. They also tried to detect similar daily behavior.

They use many DNS Answer-Based Features to detect if the IP addresses in the DNS answer look too diversified. They check the number of different IP addresses resolved for a given domain, number of different countries that these IP addresses are located in, reverse DNS query results of the returned IP addresses, number of domains that share the IP addresses that resolve to the given domain.

They also use TTL (Time To Live) Value-Based Features. For botnets to set TTL value low is important for Fast Fluxing techniques described in Section 2.2.2.3.

They extracted some domain name based lexical features assuming these being easily readable and simple to remember which is important for benign users, but not for malicious ones. They checked the ratio of numerical characters to the length of the domain name and the ratio of the length of the longest meaningful substring.

Their experimental results show that their approach works well in practice, and that it is useful in automatically identifying a wide category of malicious domains such as botnet command and control servers, phishing sites, and scam hosts.

3.5 Correlating Domain Registrations and DNS First Activity in General and for Malware

Stoner and his associates try to correlate domain registrations with the domains first DNS activity (Spring, Metcalf, & Stoner, 2011).

The main idea this paper is based on the following: behavior of malicious and average domains are different after the registration of the domain name. Registering a domain is a process, first you have to go to a registrar who will collect the necessary information and payment and then pass the new domain off to the registry. Every domain has to go through this process being malicious or benign.

For the study, they gained access to passive DNS data just like in the paper described in section 3.4. They collected 2,783,497 newly registered domains from biz, com, info, mobi, and net, top-level domains. From CERT data, they found 4729 unique domains for the malicious dataset.

During the research, they found significant difference between the first resolution of malicious and average domains. For average domains the majority of them (52.9%) were resolved within 1 day, however relatively few (4.7%) were resolved on the same day they were registered. For malicious domains, 33.2% were resolved on the same day, but observing domains registered in October, they found vast majority of domains (73%) resolved on the same day as their registrations happened.

3.6 Mining DNS for Malicious Domain Registrations

He, Zhong, Krasser, & Tang sets the challange to find malicous domains using light-weight lexical and zone based features (He, Zhong, Krasser, & Tang, 2010).

As stated in this paper, millions of new domains are registered every day, and many of them malicious. Nowadays the technique to decide if a domain is malicious or not, is to get the web content and analyze it manually or by using machine-learning techniques. However, malicious domains are often registered for only a short period of time, and there are too much of them, so manual decisions might take too long. They state: *"Due to the large numbers of new domains registered every day (many of them being potentially malicious) and the generally short lifetime for such malicious domains, it is not cost effective to use the traditional web classification methods based on content"*. They set a challenge to overcome, detecting DNS abuse without using significant amount of resource and classifying a domain without knowing its web content.

To achieve these goals they use lightweight features. They use two kinds of features, textual features, and Zone based features.

Legitimate domain names consist of meaningful English words or look alike to meaningful English words, because those are easy to remember for humans. The authors of the paper suspect that many newly registered domain names are randomly generated and meaningless strings. To detect these they use several Markov Chain models. They use some heuristic textual features too, such as length of domain name, number of digits in domain name, and so forth.

They also assume that legitimate domains will not change their name servers frequently, but many malicious domains will do so. Therefore they use such a features as the total number of name servers that ever hosted a domain, the number of domains that hosted the domain but do not host it anymore, the average/maximum/minimum lengths of name servers hosting the domain, and so forth.

Their experimental results show that their lightweight approach can detect many malicious domains with a low false-positive rate.

4 Methodology and framework

4.1 Modular system design

Why are there hundred thousands of domains registered each day, this is the main question we ask, and try to answer. Are most of the registrations really malicious, or could these registrations be coming from speculators trying to boost their parked domain set, to increase the profit? I will introduce a system designed to answer these questions, with the capability of evolving.

While framing the problem (the goal of this thesis) is quite easy, solving it is a complex challenge not accomplished up to this day. There could be **many dimensions** of categorizing domain names and also **numerous sources** and methods of obtaining information about a domain to accomplish these categorizations. We selected three dimensions for categorizing as you can see in Figure 4-1, **maliciousness** (malicious or benign domain), **passive or active** usage (passive: parked, redirected, same content on multiply sites), **topic of the page** (adult, car, insurance, blogging, IT, children, etc.). A page could be malicious, adult and active, but for example until ICANN overlooks the practice of speculation a domain can be passive (parked), benign, advertising (if the content is advertisement). There could also be passive, malicious domains just like active, benign, advertising or active, benign, news and so on.

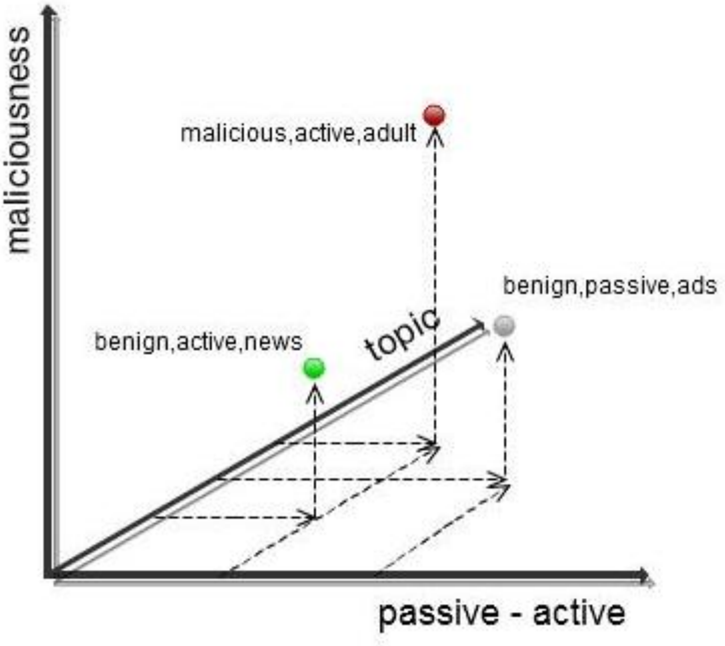


Figure 4-1, Dimensions of categorization

Categorizing along these three dimensions could bring up quite a few methodologies and we cannot foresee how well will they perform. To test multiple methodologies a framework was designed in a modular fashion. Having modules bound together in a framework has many advantages. Each module can be evaluated separately and results can be stored for other modules to work on them later or immediately. Modules built to help out other modules can be reused to assist several other modules in different researches, too. Also, the result of multiply modules, while categorizing domains, can be used together to refine results.

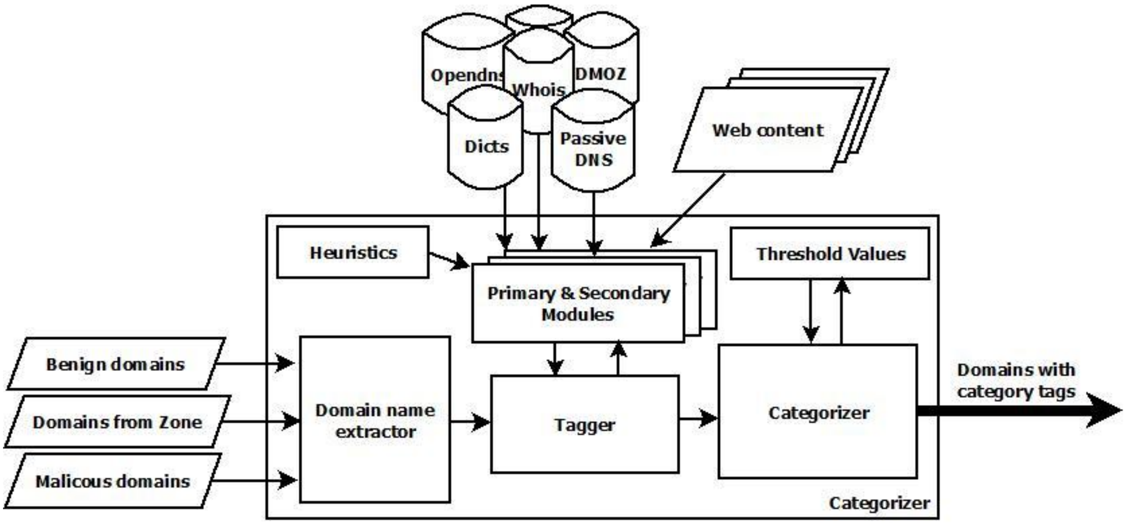


Figure 4-2, Framework architecture

Figure 4-2 shows the architecture of the framework designed, built, and used for the studies. For each test three sets of domains were used: malicious, benign, and regular zone domains (Section 4.3). We discuss the framework for obtaining data in Section 4.2. Domain name extractor module will be responsible for extracting domains from various sources, and to prepare these domains for the Tagger for further processing. When the Tagger receives the domains in an appropriate form it calls the Primary Modules. The Primary Modules are responsible for tagging domains, using the different methodologies and sources (discussed in Section 4.2). We implemented Secondary Modules to support the Primary Modules with features or services when needed. After the Tagger collected all the tags from the Primary Modules, it sends the tagged domains to the Categorizer module. The Categorizer will position the domains in the dimensions of the categories. The Categorizer is designed to categorize using heuristic thresholds, but an updated version could set threshold values in the future using machine-learning techniques as well. Finally, the output of the Categorizer is domains equipped with category tags.

4.2 Information sources on domain names

In this section I will show information sources, which can be used to categorize domain names. As there are many of them, I will present the sources in the order of how soon they are available to us after registration. It is a very important feature of an information source, because in case of speculation or malicious registration we want to conclude as soon as possible.

4.2.1 The domain name itself – lexical analysis

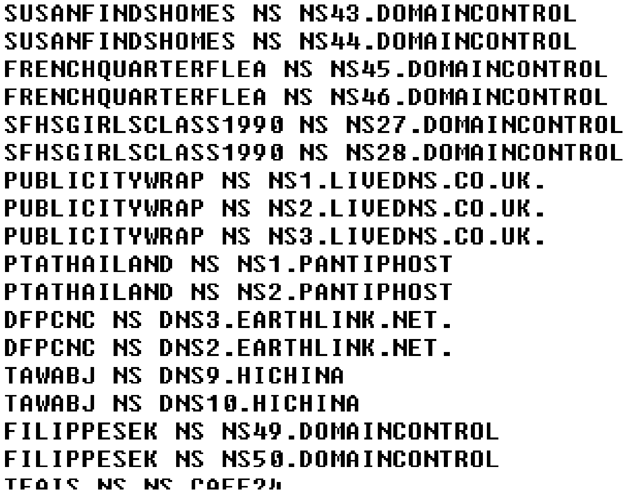
Lexical analysis of domain names is the **most lightweight** analysis that could be done, with the potential of gaining useful information about the purpose of domain registrations. It is important to be able to tell if a domain is malicious as soon as possible, because cyber criminals typically use freshly registered domains only for a short period of time (considering these domains will soon be discovered and blacklisted). Lexical analysis can be done right after the registration by retrieving new registrations from the zone file, therefore malicious domains can be discovered before they can be used for illegal activities, and this is called **proactive blacklisting**.

The drawback of lexical analysis is that we know very little, only the domain name, no usage, no content. This makes it difficult to do quality categorization, and in many cases it is impossible to do any at all. Telling if a domain is parked from lexical features is hopeless, because, as we have seen earlier in Section 2.2.1.5, it can be only established for sure by content analysis or in some cases by creating clusters based on zone file and Whois data.

For malicious purposes criminals need a lot of domains, consequently they have to generate them. Telling if a domain name was automatically generated could distinguish the good from the bad. The topic of a domain can also be decided from the domain name sometimes, for example cheappills.com is probably a pharmacy page, however carshop.com is undoubtedly about cars.

4.2.2 Zone file

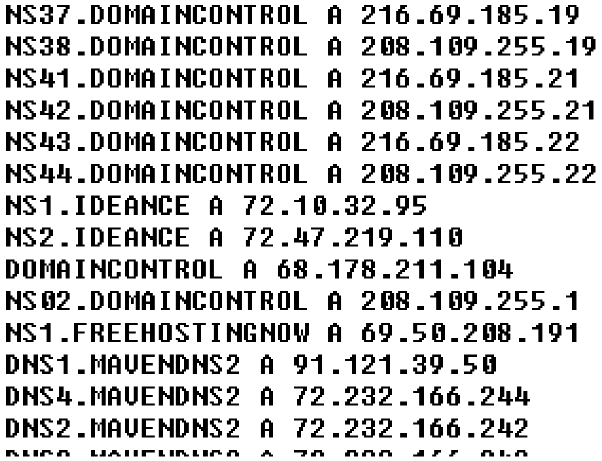
Zone files represent subsets of the hierarchical domain name structure. A DNS zone file is usually in BIND format, and contains resource records (RR). In case of a TLD zone file, the RRs are mainly name server (NS) records of second-level domains under the TLD's zone. For example, Figure 4-3 show an excerpt from the com zone file collected from VeriSign on 2011.04.14. The first line shows an NS record of susanfindshomes for we can find the IP address on ns43.domaincontrol.



```
SUSANFINDSHOMES NS NS43.DOMAINCONTROL
SUSANFINDSHOMES NS NS44.DOMAINCONTROL
FRENCHQUARTERFLEA NS NS45.DOMAINCONTROL
FRENCHQUARTERFLEA NS NS46.DOMAINCONTROL
SFHSGIRLSCLASS1990 NS NS27.DOMAINCONTROL
SFHSGIRLSCLASS1990 NS NS28.DOMAINCONTROL
PUBLICITYWRAP NS NS1.LIVEDNS.CO.UK.
PUBLICITYWRAP NS NS2.LIVEDNS.CO.UK.
PUBLICITYWRAP NS NS3.LIVEDNS.CO.UK.
PTATHAILAND NS NS1.PANTIPHOST
PTATHAILAND NS NS2.PANTIPHOST
DFPCNC NS DNS3.EARTHLINK.NET.
DFPCNC NS DNS2.EARTHLINK.NET.
TAWABJ NS DNS9.HICHINA
TAWABJ NS DNS10.HICHINA
FILIPPESEK NS NS49.DOMAINCONTROL
FILIPPESEK NS NS50.DOMAINCONTROL
TEAIS NS NS CAFE24
```

Figure 4-3, 2011.0414 com zone file excerpt 1 (from VeriSign)

The IP address of ns43.domaincontrol NS can be found also in the zone file. Records A contain the mapping between a domain and an IP address. We can see on Figure 4-4 that the IP address of ns43.domaincontrol is 216.69.185.22. If we are looking for the IP address of susanfindshomes.com, we will get back the IP address of ns43.domaincontrol and we can ask ns43.domaincontrol for the IP of susanfindshomes.com.



```
NS37.DOMAINCONTROL A 216.69.185.19
NS38.DOMAINCONTROL A 208.109.255.19
NS41.DOMAINCONTROL A 216.69.185.21
NS42.DOMAINCONTROL A 208.109.255.21
NS43.DOMAINCONTROL A 216.69.185.22
NS44.DOMAINCONTROL A 208.109.255.22
NS1.IDEANCE A 72.10.32.95
NS2.IDEANCE A 72.47.219.110
DOMAINCONTROL A 68.178.211.104
NS02.DOMAINCONTROL A 208.109.255.1
NS1.FREEHOSTINGNOW A 69.50.208.191
DNS1.MAVENDNS2 A 91.121.39.50
DNS4.MAVENDNS2 A 72.232.166.244
DNS2.MAVENDNS2 A 72.232.166.242
```

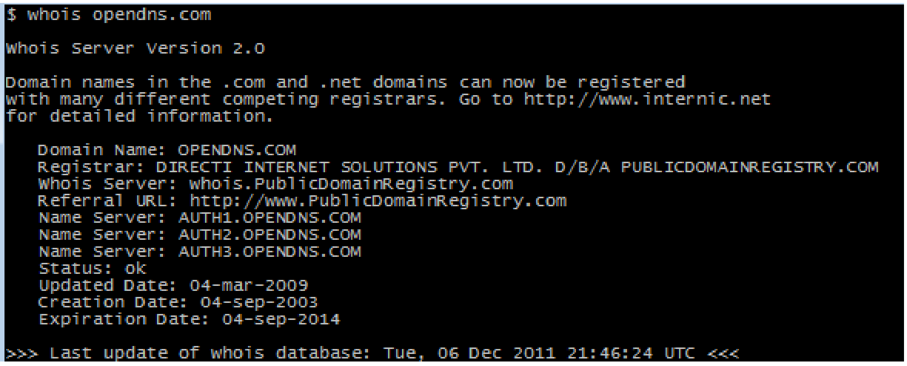
Figure 4-4, 2011.0414 com zone file excerpt 2 (from VeriSign)

TLD zone files provide us the list of the second-level domains under the TLD, but also give us name server information which can be used for clustering techniques utilized in Pro-Active Black listing proposed by Felegyhazi, Kreibich, & Paxson (Felegyhazi, Kreibich, & Paxson, 2010). We can employ it for DNS based research, for example DNS probing used in Chapter 6.

4.2.3 Whois information

Whois database contains registration information about the domain name's registrar and registrant. I will show you the data contained in a Whois record by the example of opendns.com's Whois record. How does Whois data look, how it is formatted and stored, and what it contains is determined by the TLD and is different nearly for all TLDs. In case of com maintained by VeriSign, there are two kinds of Whois data: thin Whois and thick Whois.

Thin Whois data contains information about the registrar of the domain and contains some basic information: the list of NSs, creation date, expiration date, Whois server and so on. As we can see in Figure 4-5 opendns.com has three NSs (auth1.opendns.com, auth2.opendns.com, auth3.opendns.com), it was registered on the 4th of September 2003, and registration will expire on the 4th of September 2014. Thin Whois data is standardized for each registrar.

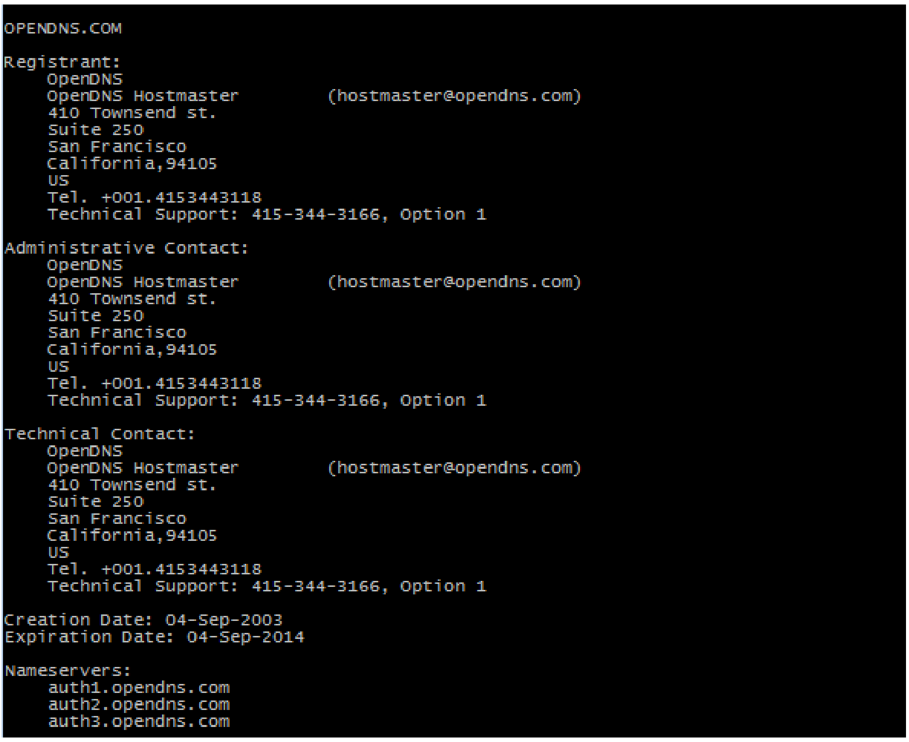


```
$ whois opendns.com

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

   Domain Name: OPENDNS.COM
   Registrar: DIRECTI INTERNET SOLUTIONS PVT. LTD. D/B/A PUBLICDOMAINREGISTRY.COM
   Whois Server: whois.PublicDomainRegistry.com
   Referral URL: http://www.PublicDomainRegistry.com
   Name Server: AUTH1.OPENDNS.COM
   Name Server: AUTH2.OPENDNS.COM
   Name Server: AUTH3.OPENDNS.COM
   Status: ok
   Updated Date: 04-mar-2009
   Creation Date: 04-sep-2003
   Expiration Date: 04-sep-2014

>>> Last update of whois database: Tue, 06 Dec 2011 21:46:24 UTC <<<
```

Figure 4-5, thin Whois record example

Thick Whois is provided by the registrar of the domain and is not standardized. It will look different for each registrar. Thick Whois data contains information about the registrant of the domain name. You can see in Figure 4-6 the information provided about opendns.com. There is the name, address, phone number, e-mail address of the registrant. You can also find administrative and technical contact details. Here the NS information, creation date, and expiration date is repeated.

This information can be used for clustering. Also, Whois data is a good addition to NS based clustering using registrant's information. It can be used for pro-active black listing mentiond previously (Felegyhazi, Kreibich, & Paxson, 2010).



```
OPENDNS.COM

Registrant:
    OpenDNS
    OpenDNS Hostmaster        (hostmaster@opendns.com)
    410 Townsend st.
    Suite 250
    San Francisco
    California,94105
    US
    Tel. +001.4153443118
    Technical Support: 415-344-3166, Option 1

Administrative Contact:
    OpenDNS
    OpenDNS Hostmaster        (hostmaster@opendns.com)
    410 Townsend st.
    Suite 250
    San Francisco
    California,94105
    US
    Tel. +001.4153443118
    Technical Support: 415-344-3166, Option 1

Technical Contact:
    OpenDNS
    OpenDNS Hostmaster        (hostmaster@opendns.com)
    410 Townsend st.
    Suite 250
    San Francisco
    California,94105
    US
    Tel. +001.4153443118
    Technical Support: 415-344-3166, Option 1

Creation Date: 04-Sep-2003
Expiration Date: 04-Sep-2014

Nameservers:
    auth1.opendns.com
    auth2.opendns.com
    auth3.opendns.com
```

Figure 4-6, thick Whois record example

4.2.4 Content analysis

Content analysis is one of the most effective ways of categorizing a domain. Whether a domain is in passive or active usage can only be decided for sure by parsing the content. If the domain is not redirected and the content is not advertisement or the domain is not for sale, then it probably has an active content.

It is also possible to create tag clouds from known sets typical for a topic, and check the content of the newly registered domains for those tags. This technique can be used on the domain name itself, but much more effective when used on the content.

4.2.5 Passive DNS data

Passive DNS data consists of DNS queries going through DNS servers on the Internet. Near real-time data can be, collected from high volume DNS sensors located at several places on the Internet. This data can be used to distinguish if malicious domains get resolved differently than benign ones. As we have seen in recent studies described in Section 3.4 and 3.5 passive DNS data is an effective tool to differentiate malicious usage of the DNS from benign.

4.2.6 Online directories

Online directories categorizing webpages, like DMOZ, Yahoo directory, and OpenDNS could be a useful help. We can simply search them for a domain we want to categorize and see how they categorize it. They are highly reliable (if their policy is strict enough) as they use the human Internet community or their staff to position domains in categories. One drawback of online directories is that categorizing domains takes them a while, therefore newly registered domain names won’t be there normally, nor is their dataset comprehensive enough for categorizing older domains, because they simply don’t contain enough domains, as we will see later, in Section 4.3.3.

However, even if we can’t use them for direct categorization still we can extract information typical for a topic. Using this information, we can make tag clouds for each topic and try to categorize domains based on these tags.

In many studies DMOZ, Yahoo directory or OpenDNS are explored to gain a benign domain set that can be used for Machine learning techniques or to extract information on benign domains. In Section 4.3.3 I will also investigate how benign these domains are.

4.2.7 Blacklists

Blacklisting is one of the fundamental ways for a community on the Internet to block unwanted entities from gaining access to resources. There are IP blacklists to ban known malicious machines and domains blacklist to ban domains, which had been used in criminal activity like spamming. To describe roughly how blacklists work is quite easy: when someone (human or machine) sees a domain participating in malicious activity, he reports the domain and after verification that domain gets blacklisted. The biggest drawback of blacklists is that, they only stop malicious activity when it has already been done. Usually cyber criminals use IP addresses and domain names for a short time to protect themselves against blacklisting, more thoroughly discussed in Section 2.2.2.

In a lot of studies (some in Chapter 3), just like in this thesis, blacklists are used to acquire a sample of malicious domains. This set can then be used for machine learning techniques and for false positive checks as well.

4.2.8 Comparison of the information sources

The primary mitigation technique in the fight against miscreants is to stop their malicious activity as soon as possible. In Figure 4-7 I show information sources based on how far in time from the registration they can be employed, and how many dimensions of categorization they can be used for. Right after the registration we have three sets of information on the domain: the domain name itself (Section 4.2.1), zone file information (Section 4.2.2), and Whois data (Section 4.2.3). These pieces of information are at our service before the domain actually gets used. It makes possible pro-active blacklisting, where we blacklist domains before they are actually operating and abusing the DNS system. Another important property of these three sets of information is being lightweight. We don’t need to crawl the domain for the content, we don’t have to build a passive DNS data overtime from the DNS queries, and we don’t need human resource as in open directories and in some blacklists (human categorization being the slowest of all). I have outlined why this information is good for pro-active blacklisting of malicious domains, but it doesn’t mean they cannot be used for categorizing domains on the passive-active or topic axes as we can see on Figure 4-7.

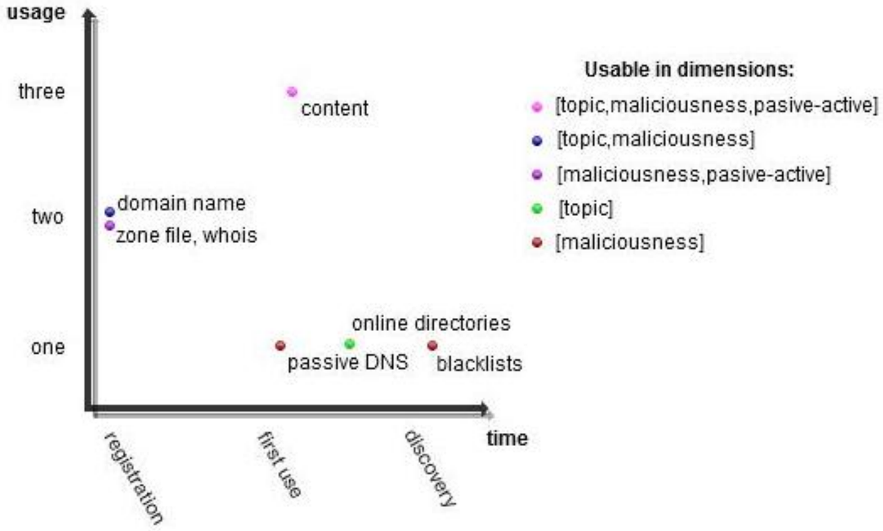


Figure 4-7, domain sources

After registration, the registrant or registrar is usually going to put content on the domain. Because of the large number of new registrations daily, it is hard to download and analyze the content. Content analysis (Section 4.2.4) is the best way to tell whether a domain is in an active or passive use. Sometimes we can find out if a domain is parked through clustering techniques, but to make sure content analysis is the reliable method. Topic of a domain cannot be concluded all the time based on the domain name, to be able to categorize all domains we need something more, the content behind. Content analysis might not be lightweight, but sometimes when we need to select the topic, or whether the domain is passive, we have to crawl for the content.

Utilizing passive DNS data (Section 4.2.5) is a heavyweight feature and can be employed, right at the first time a domain is used, to find malicious activity. It might be an effective way for telling apart malicious and benign usage, but it is still late, as recognition comes only after the DNS system has been already abused. Domains get blacklisted after the discovery of malicious activity, making blacklists often too late to be useful in stopping miscreants (but better later then never). Open directories use the Internet community to categorize domains. It makes them reliable, but it can also take a very long time for a domain to reach the directory after registration.

4.3 Domain sets used for analysis

4.3.1 Regular domain sets

Zone files are used as the source of ordinary domain sets. By regular, I mean domains we don’t know anything about in advance, and they could equally be malicious or benign (so this is a mixed dataset that has not been pre-filtered according to maliciousness). For both studies (Chapter 5 and 6) two million com domains were selected uniform randomly out of the more than 90 million domains of the com zone file dated 2011.06.27. This is used as a regular domain dataset.

For the study discussed in Chapter 5 we also used the list of domains registered in 2011.06.27 to see how much registrations were differing from domains that have been there for a while.

For the study on the biz gTLD discussed in Chapter 6 we collected domains from the biz zone of 2011.06.27 as well alongside with the corresponding second-level domains from com zone file.

4.3.2 Malicious domain sets

Malicious domains are used for the research done in Chapter 5. Many different blacklists were used for testing the categorizer modules. Surbl (Surbl: lists, 2011) is an aggregator of many other blacklists. Their list is built from: SpamCop’s URI reports, blacklist from SpamAssasin rule set, Outblaze URI blacklist (also spam), AbuseButler, and Joe Wein’s lists. They also include lists for phishing and malware, such as MailSecurity, PhishTank, OITC, DNS Blackhole list, Malware Patrol. From Surbl 212342 domains were selected for analysis. Three more blacklists Uribl (UriBL, 2011), Joe Wein (JoeWein, 2011), and SpamHaus (Spamhaus: DB, 2011) were used to see how results alter related to different blacklists.

4.3.3 Benign domain sets

DMOZ (DMOZ: ODP – Open Directory Project, 2011) and random Yahoo (used in many articles) were picked as benign domain sources. DMOZ is an Open Directory Project (ODP) edited by users of the web, and it is one of the most comprehensive human edited directories of web pages on the Internet, as they state it:

“The Open Directory Project is the largest, most comprehensive human-edited directory of the Web. It is constructed and maintained by a vast, global community of volunteer editors”

But even being the most comprehensive directory of the web, having 4,952,023 sites categorized by 92,857 editors in over 1,008,705 categories, DMOZ is still too small to be used for direct categorization just like OpenDNS, as we will see in Section 4.5. 429369 distinct DMOZ com second-level domain names were gathered from categories: Arts, Business, Computers, Games, Health, Home, News, Recreation, Shopping, Sports, Science and Reference.

Big advantage of DMOZ is, their data can be downloaded free as an XML file. At DMOZ anyone can become an editor choosing his or her subcategory under another already existing category. One drawback is the huge number of editors and categories, which affects the quality of domain sets in DMOZ database. Their database is uneven and some categories contain much more domains then other.

Yahoo (Yahoo: random, 2011) has a service for viewing random pages picked by Yahoo. This service was used to gather over 200 000 benign domains. While DMOZ made their directory downloadable, Yahoo can only be reached through their random service. The problem is that Yahoo does not generate a new random every time the random generator is called, only generates a new random every given period of time. For this reason approximately one domain can be downloaded in one second making very slow to get a comprehensive set of Yahoo domains. Being so slow, it didn’t matter if only the domain name were captured or if the content was downloaded too.

afiwi - Exploit
multecpkging - Trojan
flagplanet - Exploit
pbalkcom - Exploit
clearanceac - Trojan
festivefever - Exploit
investorsleague - Trojan
pretto - Trojan
spiritonin - Exploit
sodashop - Exploit

micklitz - Trojan
scienceofsales - Trojan
koobifora - Trojan
salsaguides - Exploit
welcometoindia - Exploit
casciocom - Trojan
nympho - Exploit
clipsemanagement - Virus
wizaustin - Exploit
bodybuildingnatural - VirTool

While downloading Yahoo domains Microsoft Security Essentials found some of them infected, the above list shows some examples. Totally 59 domains were found as infected during the download, 31 exploits, 21 trojans, 2 viruses, 3 virus tools and 2 trojan downloaders. Rescanning showed some more infected items. As probably Microsoft Security Essentials didn't find all infections and it was just checking the content directly on the URL http://domain.com, the question turn up, how benign really yahoo domains are?

4.4 Heuristical modules and conclusions

Heuristical analysis might not be the right choice when it comes to categorizing domain names, but it can help us foresee the properties of domains that could be used for categorizing. For testing, the two million random com domains were used from the zone file, and fed to the domain name extractor module.

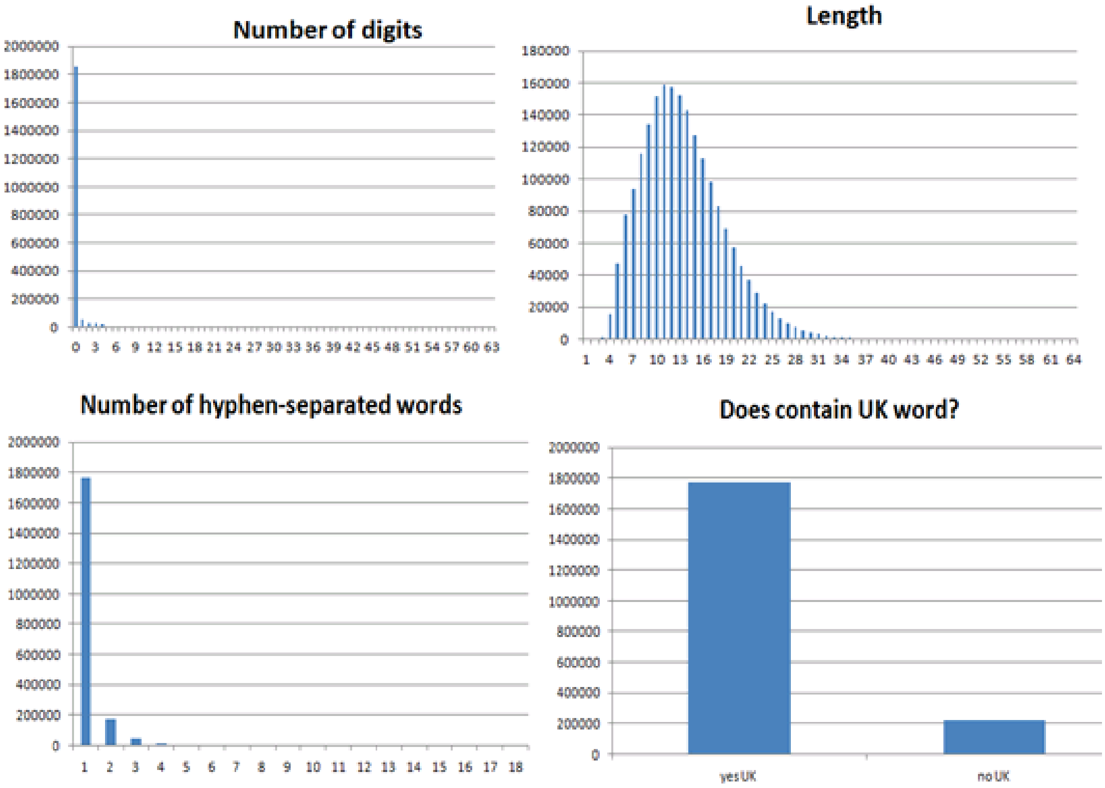


Figure 4-8, results from heuristic module

Figure 4-8 shows some results produced from the heuristic modules. Top-right diagram on Figure 4-8 shows the number of digits found in domain names. More than 92.5% of the domains do not contain any number, and 7.4% of the domains contain 1 to 11 digits. Also, many digits represent a date or stand for a word (many2many, sk8, 4you, etc.). The conclusion is, finding proof for the assumption that most domains are malicious, couldn’t be achieved by observing features based on the digits in a domain name.

A domain name can contain English letters, digits and hyphens. On Figure 4-8 substrings mean the parts of a domain coming from splitting it by the hyphens. Nearly 90% of domains do not contain a hyphen. Also, the distribution of the length of the domain names is showing a nice diagram, without any salient part. This bring us to a conclusion that neither the length of a domain, nor the number of substrings in a domain will be good starting point finding malicious or parked domains.

Finally, only 10 percent of domains do not contain any English word at all. For this measurement only one UK dictionary was used with nearly 60,000 words in it. Later tests will show us that using larger word pools, we find even fewer domains not containing any words.

Although criminals are sometimes using name generator algorithms, they want to make their domains look like regular domains as much as possible. This is probably the reason why we don’t see much hyphens or numbers in domain names, or a lot of domain in any length outstandingly. Also, this is the reason why criminals need to put words in their domains. The main conclusion of these few heuristics is that the properties of a domain analyzed in this section are probably not the right ones to establish whether a domain is malicious, parked, or benign, but they may be used to categorize a subset of the domains. What we need is a more sophisticated method, which will be shown in Chapter 5.

4.5 OpenDNS for direct categorization

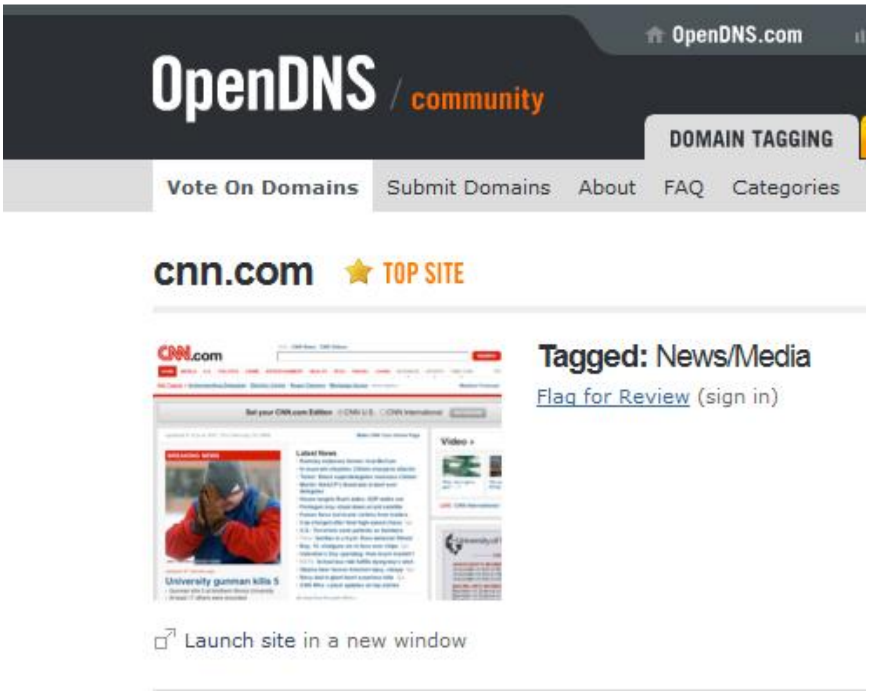
OpenDNS has a service called domain tagging (OpenDNS: Domain Tagging, 2011). They invite the users of the Internet to tag domains based on their category list. They verify the tags by relying on more users’ opinion. This service can be used for blocking unwanted categories.

OpenDNS provides an interface on their website to find out the tags they have given to a domain. For first approximation, it looked awkward to crawl their website for millions of domains. Fortunately (or unfortunately as we shall see later), we found other ways to identify the tags they given to a domain.

As OpenDNS is used for parental control, categories can be selected to block domains. OpenDNS’s DNS server can be set as the preferred and alternative DNS server for a computer, and then a queried domain is blocked if it is in the previously selected category set. When blocked OpenDNS resolves the queried domain for a page maintained by OpenDNS, which will tell us the reason of blocking.

First, a web crawler was constructed. The idea was to block all categories and try to download the page (we want to categorize). Then, if the domain is tagged in one of the categories, an OpenDNS page will be downloaded instead of the original page. Parsing the content will tell the reason for blocking: the categories of the domain. However, after this method was implemented, it proved to be too slow for testing millions of domains.

To speed things up, we relied on the idea that DNS resolving is faster than content downloading. The way to do it is: just block for one category and make a DNS query for a domain. If the DNS answer contains the IP address belonging to OpenDNS, than the queried domain was blocked meaning it is in the selected category. This way we had to make a DNS query in each category for each domain, but we hoped the speed of DNS will make up for this overhead. Sadly the DNS crawler didn't seem to be fast enough either, with an average of 0.3 s per query (For 1 million domain it would take 190 days for my laptop).



cnn.com

Tagged: News/Media

Figure 4-9: Screenshot of http://domain.opendns.com/cnn.com

After all, I had to build a web crawler specified for downloading the pages on opendns.com for domains categorized as on Figure 4-9. This method has proved to be fast enough, making categorization of Alexa’s top one million domains (Alexa: Alexa top sites, 2011) by OpenDNS possible.

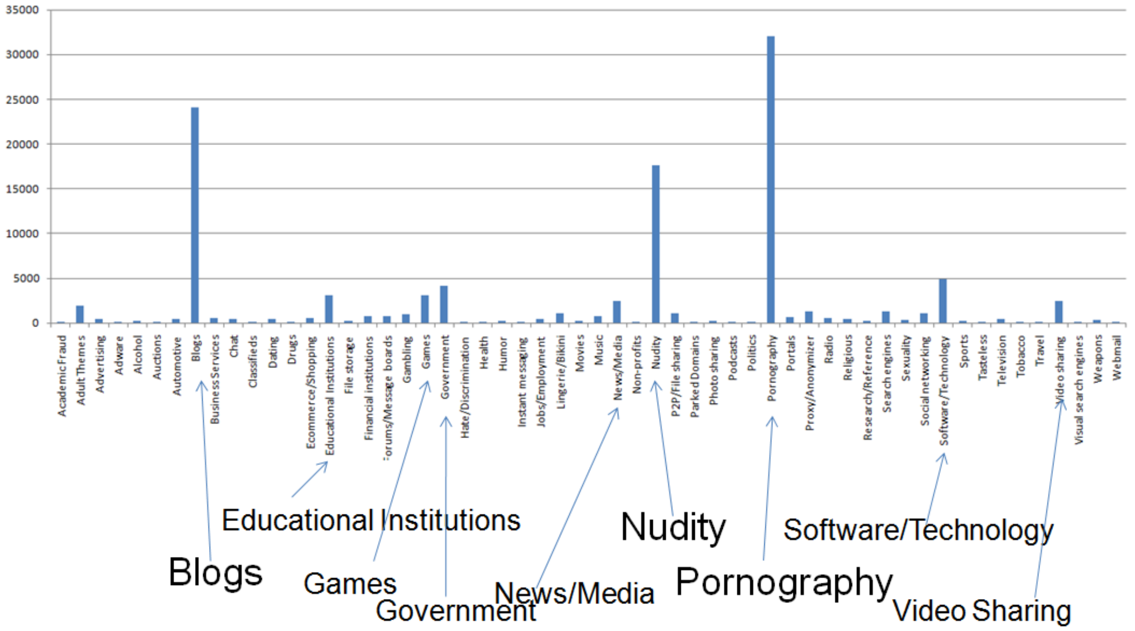


Figure 4-10 Alexa’s top 1 million categorized by OpenDNS

From Alexa’s top 1 million pages 114233 domains were categorized in the OpenDNS database. The results show on Figure 4-10 that a large fraction of domains from Alexa in the OpenDNS database contain nudity or pornography. My personal opinion is that the 43.5% of the domains being in the nudity or pornography categories doesn’t mean that 43.5% of Alexa domains are really in those categories. Probably these numbers are coming from OpenDNS’s prime focus on parental control, distorting the results.

OpenDNS has over 7.4 million domains submitted and over 1.4 million domains decided. This is less than one percent of all hundred millions of domains existing, meaning OpenDNS cannot be used for direct categorization. But it can be used for making typical domain sets for categories.

5 Usage of com domains

During the research, we first wanted to look at features available right after the registration of a domain name. Due to the complexity of the three sources at our disposal (domain name, zone file and Whois data), we restricted our attention to the first step, the lexical analysis of the domain names. Our framework allows for implementing the remaining two features and thus we leave them as future work. For the lexical analysis, we designed two features and evaluated them on the collected datasets of benign, regular and malicious domain names. As we will see later on, our investigation of the structure of domain names shows us that most of the domain names contain at least two words. We designed a module based on the content downloaded from Wikipedia (Wikipedia: random, 2011) and we present this Wiki module in Section 5.1. For domain names, containing less than two words the module introduced in Section 5.2 can be used.

5.1 Wiki Module

Because miscreants need a huge amount of domain names for their campaigns (as seen in earlier chapters), we can assume that they employ automated solutions to generate domain names algorithmically. The concept behind the Wiki Module is to find incoherent word combinations in a domain name, which will not appear in ordinary text, such as Wikipedia articles.

5.1.1 Design of the Wiki Module

Figure 5-1 shows us how the Wiki Module is designed, and how can it evaluate whether a domain is algorithmically generated or not. The Wiki Module consists of multiple primary and secondary modules.

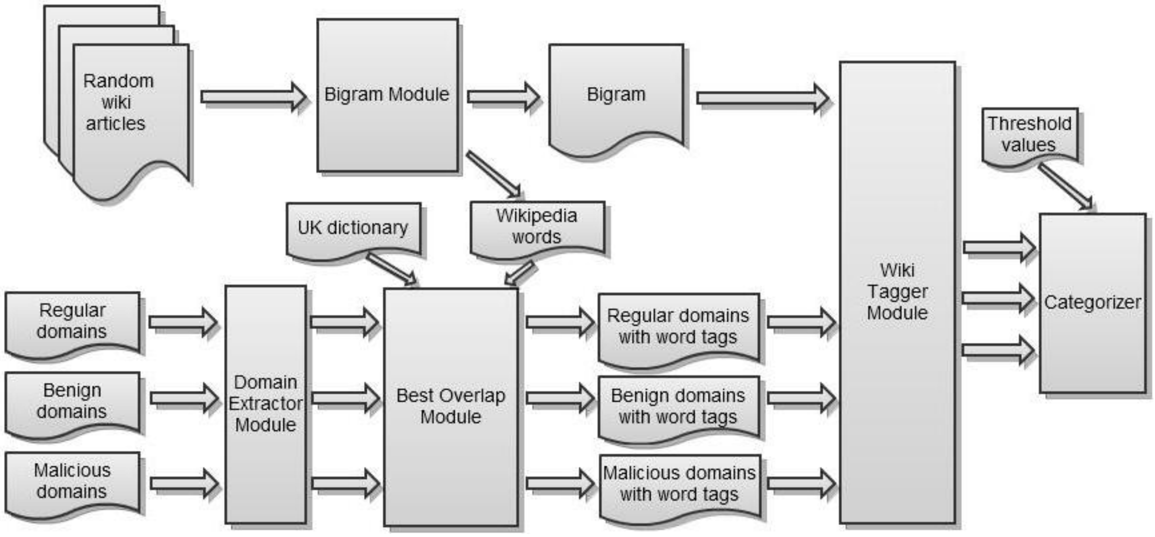


Figure 5-1, Design of the Wiki Module

If we want to find incoherent word pairs in domain names, first we have to find the words in the domain names. We do not just simply want to find all the words a domain name contains, but we want to identify potential compositions, that are non-overlapping sets of words. From these sets we want to find one, which covers the most of the domain name, this is what we call the best overlap.

First, the Domain Extractor Module acquires regular, malicious and benign domains and prepares them for the Best Overlap Module (Section 5.1.4). The Best Overlap Module will find the best overlaps (mentioned above) in these domain sets using words from UK dictionary and from downloaded Wikipedia articles.

To decide if the words are coherent in a best overlap, we are using bigrams. Bigrams tell us how frequently a word is followed by another one in a text. For creating the bigram, English Wikipedia articles were used as text. Random Wiki Crawler described in Section 5.1.2 was used for downloading Wikipedia pages. When we see a word combination with high frequency, it means the two words are coherent, but the ones not in the bigram or in the bigram with small frequency are considered incoherent. The Bigram Module is responsible for creating bigrams from the Wikipedia articles (Section 5.1.3).

In Figure 5-1 we can see that the Wiki Tagger Module (Section 5.1.5) uses the output of Bigram Module: the bigram, and the output of the Best Overlap Module. The Wiki Tagger Module’s role is to tag, how coherent are the word pairs found by the Best Overlap Module, using the bigram. To decide whether the domain name consists of incoherent words is the responsibility of the Categorizer based on the output of the Wiki Tagger Module and the threshold values.

During this research, each module will store the results in files and each module will use the other modules' files as an input. This methodology is very important as some modules need very high computing capacity and others need a lot of RAM, so we might want to run them separately. Another advantage of this design is that, we are often going to use different versions of modules trying to find the best solution, and yet doing this we still don't have to throw away the partial results we already have.

5.1.2 Random Wiki Crawler

We are going to use Wikipedia articles for three modules. The primary usage will be to make a word bigram, but we also utilize the articles for building a large dictionary and in Section 5.2 we use them to make a letter bigram.

Wikipedia has a service for reading random wiki articles. The crawler downloads these random pages, always remembering the downloaded pages therefore pages will not be downloaded twice.

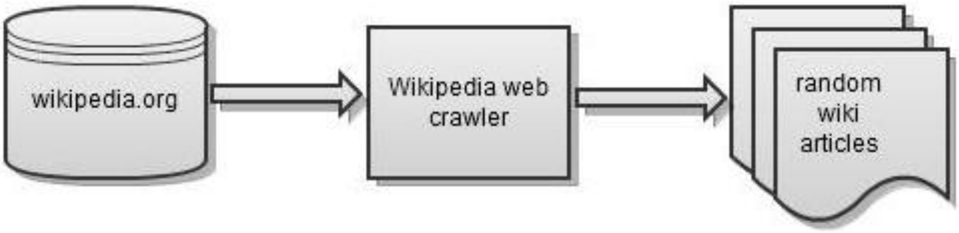


Figure 5-2, Wiki crawler

To download the Wikipedia pages most effectively, I designed a multithread crawler. When crawling the web, the response time of downloading a webpage is the bottleneck of time efficiency, not the computational speed of a server or PC. To improve this attribute of crawling, multithreading is used. In each thread we start a download separately, therefore we can utilize the maximum capacity of our Internet connection. The number of threads is adjustable, but during downloading Wikipedia pages I never set it to high, not to put too much load on Wikipedia's servers.

To gain a comprehensive set of text for lexical analysis from Wikipedia we need hundred thousands or millions of articles. Earlier experiences show that having a lot of files in a directory using Linux system will cause the response time of reaching a file increase, resulting in the non-usability of any software working with these files. The solution is to use a hierarchical directory structure to store files. We created a folder for each possible first letter for Wikipedia article names and put the articles in the corresponding directories, see Figure 5-3.

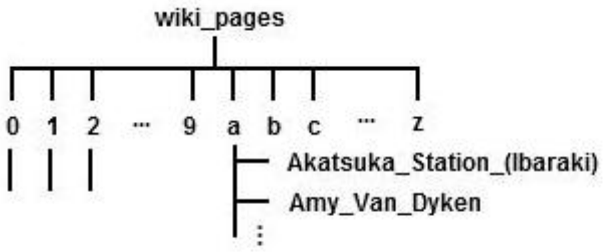


Figure 5-3, directory structure for Wikipedia articles

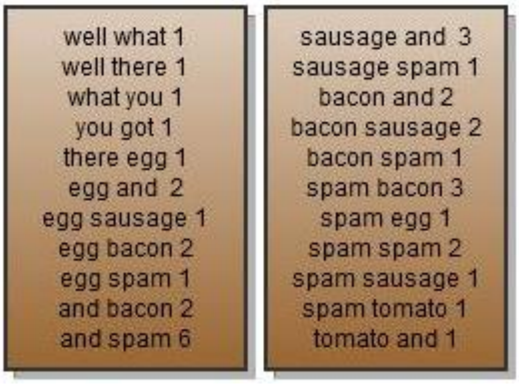
5.1.3 Bigram Module

The Bigram Module is responsible to find relations between words. To achieve its goal the Bigram Module parses each Wikipedia articles for paragraphs in them, and breaks the paragraphs to sentences. From each sentence it collects the word pairs found in them. Let us follow through an example how Bigram Module finds word pairs in text. For the example, I selected a part from the famous Monty Python sketch titled “Spam”:

“Man: Well, what’ve you got?

Waitress: Well, there’s egg and bacon; egg sausage and bacon; egg and spam; egg bacon and spam; egg bacon sausage and spam; spam bacon sausage and spam; spam egg spam spam bacon and spam; spam sausage spam spam bacon spam tomato and spam;”

The result is shown in Figure 5-4:



well what 1
well there 1
what you 1
you got 1
there egg 1
egg and 2
egg sausage 1
egg bacon 2
egg spam 1
and bacon 2
and spam 6

sausage and 3
sausage spam 1
bacon and 2
bacon sausage 2
bacon spam 1
spam bacon 3
spam egg 1
spam spam 2
spam sausage 1
spam tomato 1
tomato and 1

Figure 5-4, Bigram "Spam" example

At first, we looked for one-way relations. The word “egg” is followed by “bacon” two times, but we never see the inverse of it. Accordingly, if we will look at the domain baconegg.com we will not find it related. Later in the study, we switch to two-way relations to improve our categorization.

5.1.4 Best Overlap Module

For the Wiki Module we want to find the best overlap (described in Section 5.1.1) of words in domain names. The problem of finding best overlaps is that we observed it couldn't be calculated in polynomial time. The consequence being when a domain name is too long and contains a lot of words the algorithm won't be able to finish in time. To avoid this problem, in case of very long domain names, which are not very frequent, we use only some of the longest words found in the domain name.

An example for the Best Overlap module is the domain templatekingdom.com found in the DMOZ set. The templatekingdom.com domain name contains many words: template, kingdom, plate, king, temp, late, plat, lat, tem, kin, ate and so on. Our algorithm will find in this case the words template and kingdom, but an equally long set would be temp, late and kingdom. Note that the algorithm does not guarantee a unique result. Thus, a future improvement could be that we save the first couple of best overlaps, and use the one, which got the best score in Wiki, because this is the word combination most likely to be what the registrant meant when registering the domain. Also, this improvement is safe because it just lowers the number of false positives.

5.1.5 Wiki Tagger Module

The Wiki Tagger will give the tags to the domain names, which we can use to evaluate if domains were automatically generated or not. Using the output of the Best Overlap Module, it receives the word combinations in the analyzed domain. Then it tries to find each word combination in the bigram (output of the Bigram Module). When it finds the word combination it logs the frequency, else it logs not found. Also, the sum of the frequencies and the number of not found word combinations are logged.

Best Overlap module does not log the order of words found, therefore I had to create an algorithm for finding order of words in a domain name. The tagger works very fast. The algorithm will try to find the first word in the list. When found it, splits the domain name by the word just found and for further processing it uses the second half of the domain name. Then it tries to find the second word and if found it, splits the domain name by the second word and uses what is left of the domain name for further processing. It goes on like this, till it does not find a word. When it does not find a word, it will mean the word was before the one we found last time. The algorithm than changes the places of the last found word and the word not found. Our example will be studylinkonlinetutoring.com and the words found are tutoring, online, study and link:

1. tutoring, online, study, link   found tutoring; not found online
2. online, tutoring, study, link   found online, tutoring; not found study
3. online, study, tutoring, link   found online; not found study
4. study, online, tutoring, link   found study, online, tutoring; not found link
5. study, online, link, tutoring   found study, online; not found link
6. study, link, online, tutoring   found study, online, tutoring and link

By the sixth step, it has found the order of words in the domain name. This algorithm nearly always finds the order of words, but in some cases it gets stuck. Try to run the algorithm for imaginary domain bluecatblues (words are cat, blues and blue):

1. cat, blues, blue   found cat, blues; not found blue
2. cat, blue, blues   found cat, blue; not found blues
3. cat, blues, blue   found cat, blues; not found blue

We can see that the list of words in the first phase is the same as in the second phase. Because it happens only in a few cases, I used a simple solution. If my algorithm gets stuck, I just simply start to reorder the word list randomly. During the research it always proved to be real fast and even running millions of domain names the Tagger has finished in a couple of minutes.

5.1.6 Analysis

For the first approximation, the previously mentioned UK word list was used for finding words and the best overlap in domains. The first tests were run on three sets of domains: half million random com domain, 429369 DMOZ domains and 212342 Surbl domains. The results are shown below in Figure 5-5.

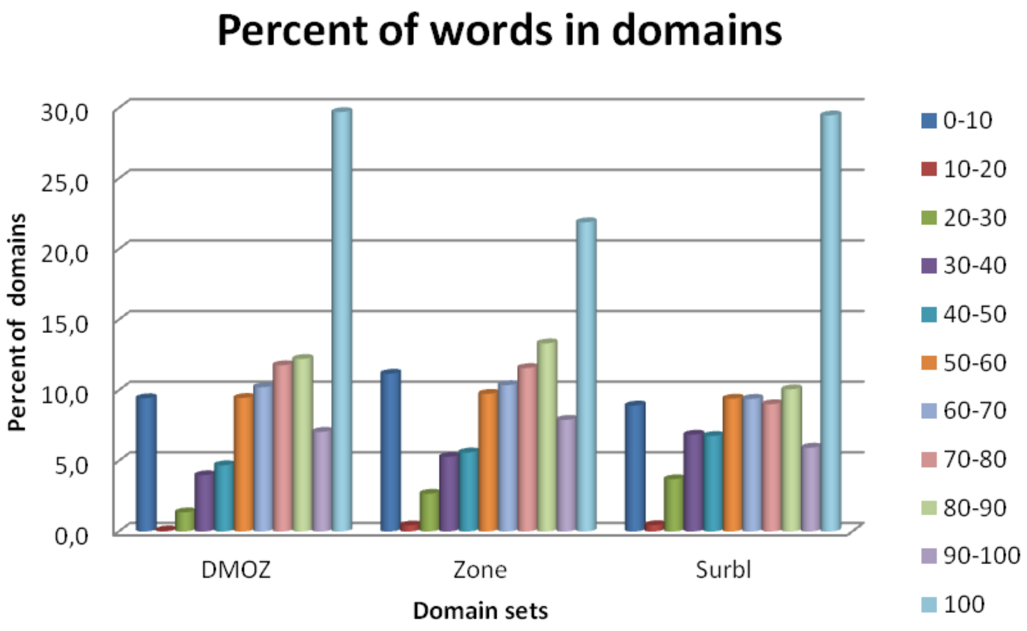


Figure 5-5, Best Overlap Module's results using UK words

Figure 5-5 shows the percent of words in a domain name. For example, the orange column above DMOZ means that around 10 percent of DMOZ domain names contain words in 50 to 60 percent of their total length. The dark blue columns nearly always mean domains which don't contain any words (0-10% tends to only be 0%), because it happens only in a very few cases that a domain name contains a word and it is less than 10 percent of the length of the domain name.

For DMOZ only 9.4%, for Zone only 11.2% and for Surbl only 8.9% of domain names do not contain any words. In fact respectively 80.5%, 74.8% and 73.3% of domain names possess more than 50% of words, this shows us that finding relations between words in a domain name can be used to analyze most of the domains.

More precisely Wiki Module can only be used on domains that contain at least two words. Figure 5-6 shows as how many domains contain zero word, one words or more than one word. It shows that in DMOZ, Zone and Surbl respectively 63.5%, 65.2% and 67.7% of domain names contain at least two words, making them available for analysis by the Wiki Module

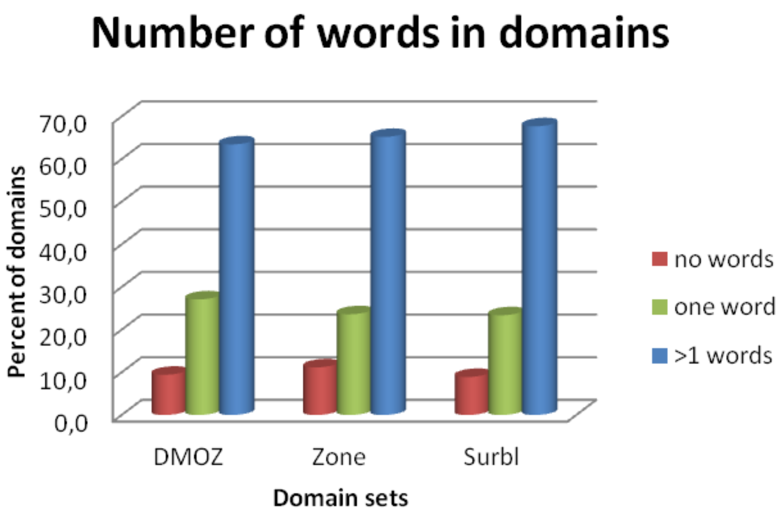


Figure 5-6, Number of words in domains using UK word list

While running the Best Overlap Module, using the Wikipedia crawler, 200,000 random English articles have been downloaded. From these articles, the Bigram Module built the first bigram containing nearly 7.5 million relations. Using the output of the Bigram Module and the Best Overlap Module, the Wiki Tagger Module gave the results shown in Figure 5-7.

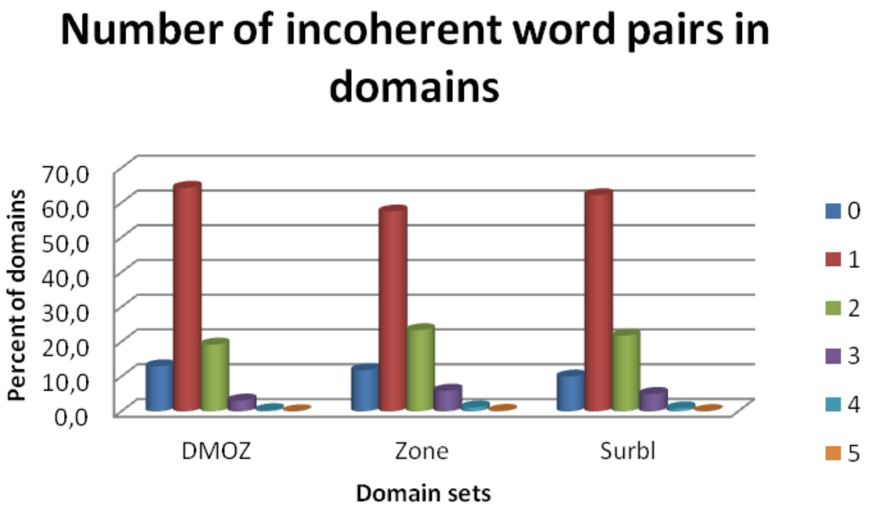


Figure 5-7, Wiki Tagger Module results using UK word list

Figure 5-7 shows how much incoherent word pairs were found in domain names (considering domains containing more then on word). 10% Surbl, 11.8% Zone and 13% DMOZ domains contain only coherent word pairs (zero incoherent). While it seems that DMOZ has more domains containing zero incoherent word pairs then Zone and Surbl, this 3 percent difference is not significant. Approximately 70% of each domain sets have received 0 point, which means in those domains no coherent word pairs have been found. The conclusion of results in Figure 5-7 is that our method couldn't discover most of the relations between words, consequently it is not ready to find algorithmically generated domain names.

There are two main reasons why the first run of Wiki Module has performed poorly. The first reason is: using one UK dictionary with about 60000 words in it is not comprehensive. It does not contain words like personal names, geographical names, corporation names, abbreviations and so on, and these words are commonly used in domain names. The other main reason is: we probably do not have enough word pair relations in our bigram.

To acquire a more comprehensive word list, we subtracted all words found in the over 200000 Wikipedia articles, and from the 953094 most frequently occurring words nearly 150000 were selected (and added to the UK word list).

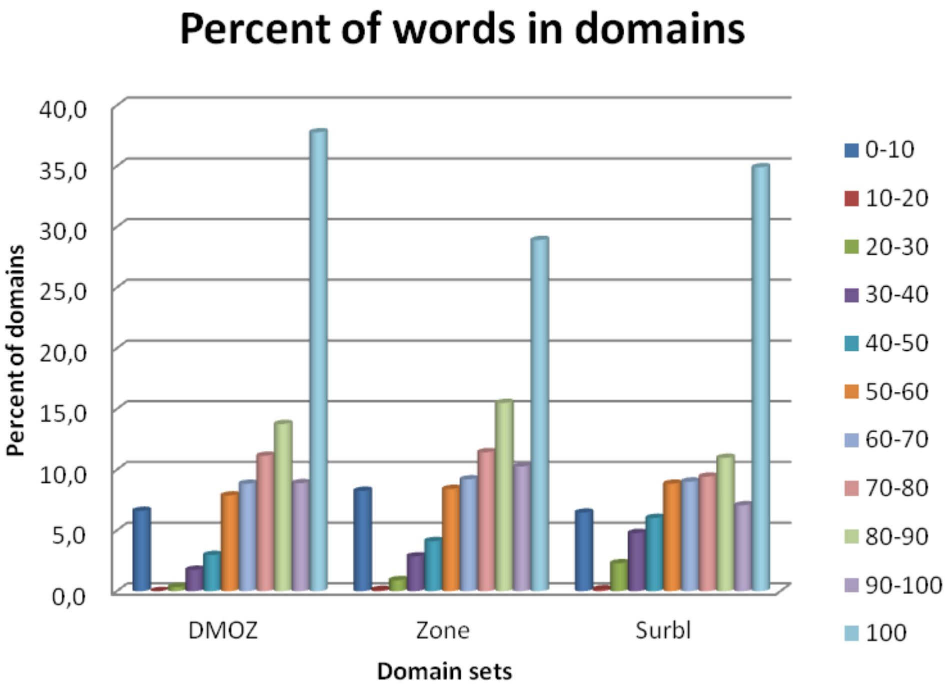


Figure 5-8, Best Overlap Module's results using UK + Wiki200k word lists

In Figure 5-8 we can see the results of the Best Overlap Module using the new improved word list. As expected the number of domains not containing words (dark blue column) decreased 3% compared to using just UK word list. Also, the number of domains consisting of words only (light blue column) has increased by 9 to 5 percent (DMOZ and Surbl). 88.3% of Surbl, 83.7% of Zone and 80.3% of DMOZ domain names' have more than 50% words in them. For each set approximately 70% contains at least two words, which is a 5% improvement.

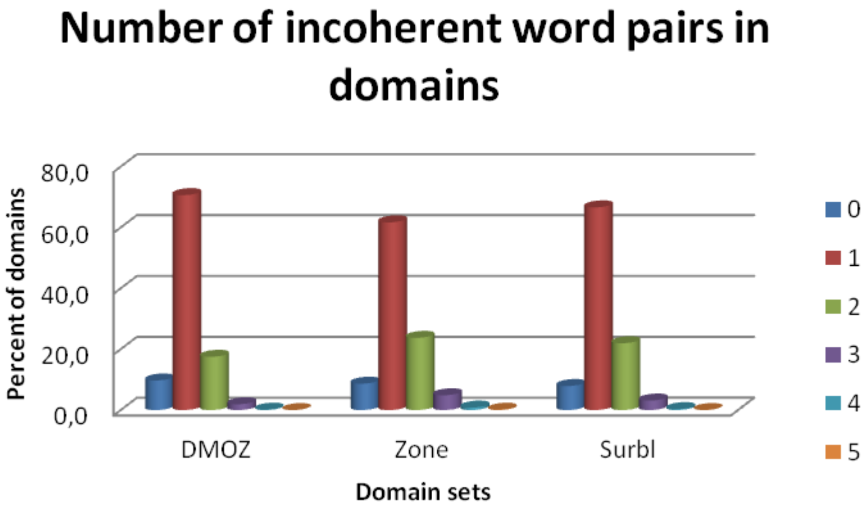


Figure 5-9, Wiki Tagger Module results using UK + Wiki200k word lists

Sadly, Figure 5-9 shows no improvement, because we found only more words in the domain names, but eventually did not find more relations between words and the results even declined a bit.

To improve on the number of relations I have lightened the strict definition of bigram so it could contain word pairs not related directly. For example, consider the sentence: "Blue dogs like red cats". Original bigram would have words pairs like: blue-dogs, dogs-like, like-red, red-cats. But our lightened bigram will pair words positioned maximum three places apart from each other: blue-dogs, blue-like, blue-red, but no blue-cats (have you ever seen a blue cat?). This bigram will have word pairs directly not related, for example: blue-like. Also, it is not impossible that for someone the domain name bluelike.com (it is actually a parked domain) is meaningful. Now it doesn't really matter anymore, because of our approach. The policy is: not finding a generated domain is not as bad as marking a domain generated when it is not. Subsequently we are set to find as many relations as possible.

Many improvements have been made in this thesis for the last analysis. The Wikipedia has been crawled again to collect over 1.1 million English Wikipedia articles nearly 30% of all articles. We found over 1.9 million words from this many articles.

To make a more comprehensive word list we selected the most frequent words from the articles. We used a rule: the shorter the word is, the more frequent it has to be, to be in the word list. Also, I blended this word list with the previous UK dictionary. The word list has ended up containing 580349 words.

For the improved tests more domain sets were used, too. Yahoo was added to DMOZ as a benign domain set. From the com zone file, all of the 2 million random domains were used. We tested the newly registered domains too, as we suspected they would contain lots of malicious domains. Surbl is an aggregate of several lists, therefore domains can be in their list for many reason. Nevertheless, we also wanted to test for more specific blacklists, targeting spamming, like: JoeWein, Uribl and SpamHaus.

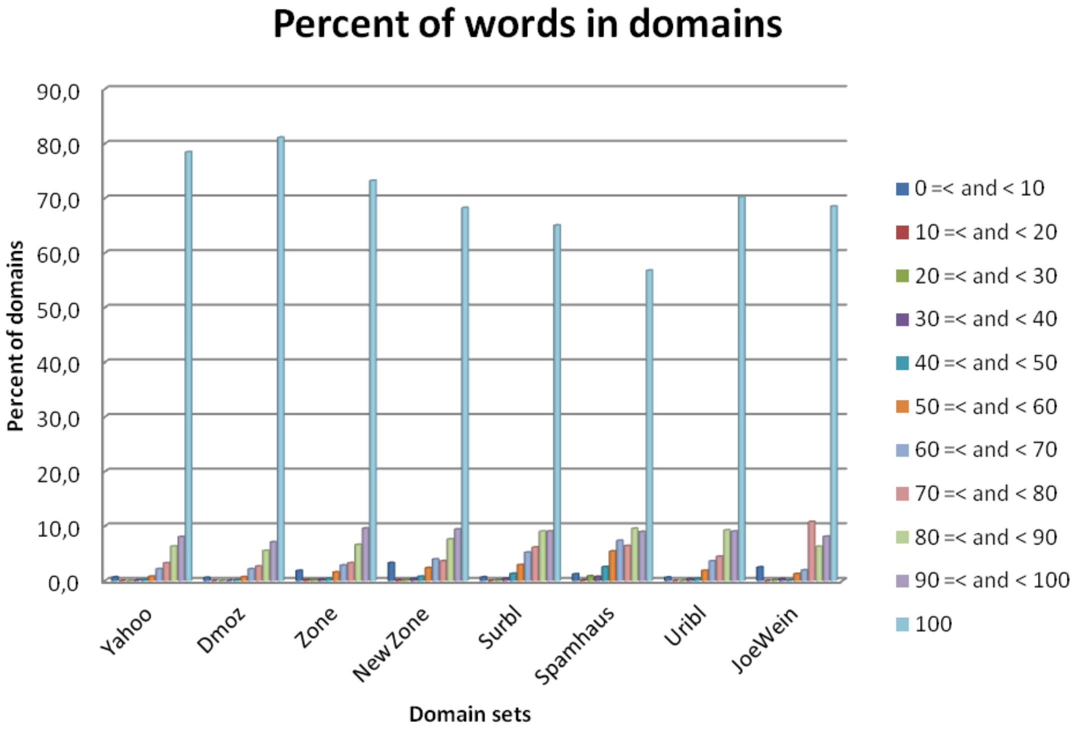


Figure 5-10, Best Overlap Module's results using Wiki 1.1 million word list

Figure 5-10 shows that crawling 1.1 million Wikipedia pages paid out. DMOZ and Yahoo have the most domains (over 99%) containing more than 50% words. However, even the last one, which is SpamHaus, has lots of domains (94.5%) containing more than 50% words. Also, the percent of domains consisting 100% of words is very high, for DMOZ 81.1%, Yahoo 78.5%, com zone 73.2% and for the new registrations 68%. For the bad domains it is lower, but still quite high Surbl 65.1%, SpamHaus 56.8%, Uribl 70.3% and JoeWein 68.5%.

These results from the Best Overlap Module support our statement that miscreants mostly pay attention to register domains containing words and look like benign domains (they do not want to register domains, which look evidently meaningless and random).

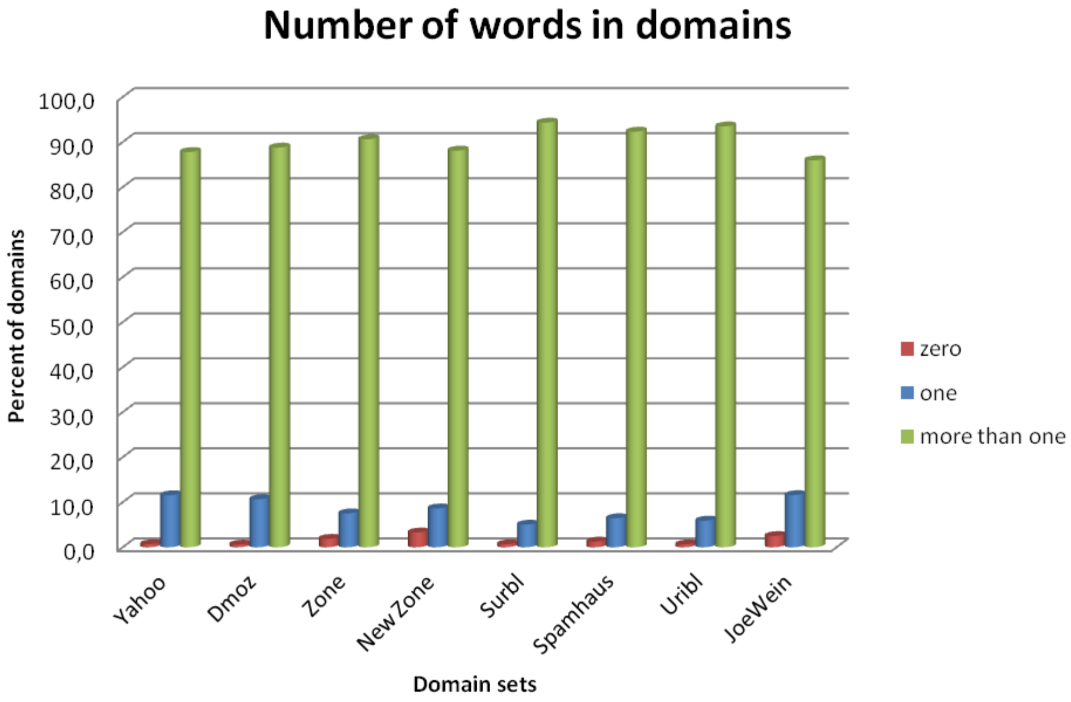


Figure 5-11, Number of words in domains using Wiki 1.1 million word list

To evaluate precisely for how much domains we can use the Wiki Module, I counted the number of words in domains. As we can see in Figure 5-11, the Wiki Module can evaluate 90% of the domains and this is true for all domain sets. Weirdly, some of the blacklists lead, Surbl is the first with 94.3% of domains containing at least two words, but Uribl is the second with 93.5% and SpamHaus is the third with 92.3%. The two lowest are JoeWein with 85.9% and Yahoo with 87.8%.

In the study (He, Zhong, Krasser, & Tang, 2010) we have discussed in Section 3.6 they state:

"One interesting pattern in legitimate domain names is that many of them consist of English words or look like meaningful English while many malicious domain names are randomly generated and do not include meaningful words. We show that it is possible to transform this intuitive observation into statistically informative features using second order Markov models."

"The experimental results demonstrate that this very light-weight approach can detect many malicious domains with a low FP rate."

My results are contradicting their statement as we can see equally high number of domains containing a lot of words for malicious and for benign domain sets. Also not surprisingly, only less than one percent of domains didn't contain any word in the Yahoo and DMOZ sets, but the random Zone set fared 1.9% too. They were trying to find domains with meaningless strings, but my analysis shows that the highest values are: JoeWein with 2.5% and new registrations with 3.3% of domain names not containing any words. This statement is contradictory with their statement: "*can detect many malicious domains*".

We had a great improvement of finding words in domain names. But using 1.1 million English Wikipedia articles improved the bigram too. Creating the new bigram with lightened relations, we have found 114450585 related words. This is over 15 times more then what we found at first try.

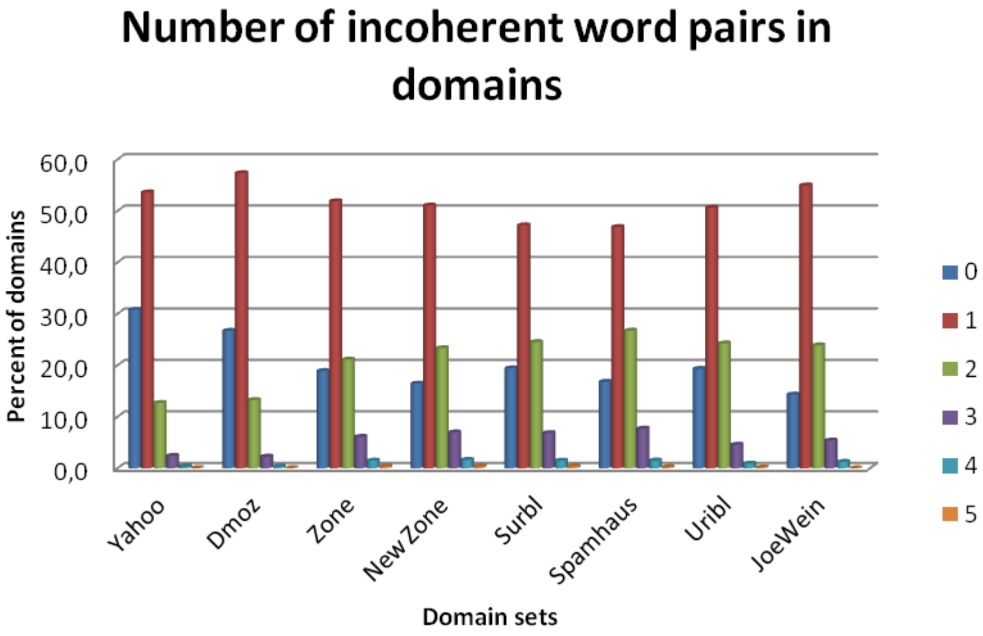


Figure 5-12, Wiki Tagger Module results using Wiki 1.1 million word list

In Figure 5-12 we can see that the percent of domains not containing incoherent words is increased for DMOZ from 9.7% to 26.7%, for Zone from 8.7% to 18.9% and for Surbl 7.9% to 19.4%. It does not just show a huge increase in finding coherent word pairs, but shows us that in case of DMOZ the ratio of improvement is significantly higher than for the Zone or the Surbl set.

To see the difference between the benign and the malicious sets Figure 5-13 takes a closer look at the results:

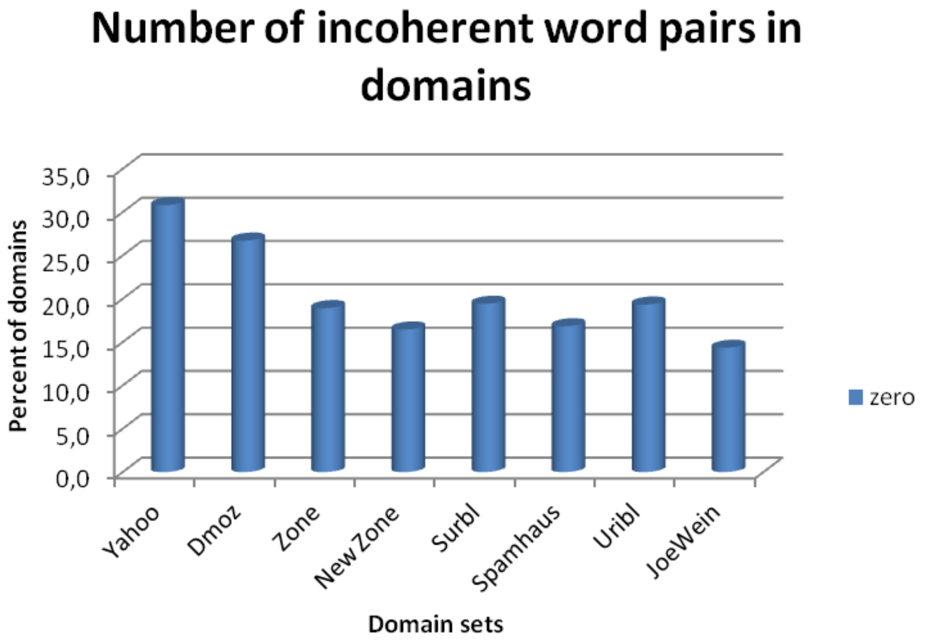


Figure 5-13, Percent of domains containing zero incoherent word pairs

Looking at the benign domain sets in Figure 5-13 we can see that 30.9% percent of Yahoo domain names and 26.7% of DMOZ domain names are built up only from coherent words. However, the results show significant difference for the malicious sets: 19.4% of Surbl, 16.8% of SpamHaus, 19.3% of Uribl, 14.3% of JoeWein domains have zero incoherent word pairs in them. Benign sets can have twice as much domain names not algorithmically generated than malicious sets. This means our method can be used to differentiate an automatically generated malicious domain set from a manually created benign set.

Another interesting consequence is that looking at Figure 5-12 or Figure 5-13 random com zone domains and newly registered domains look very similar to malicious domains. Merely 18.9% of random com domain names contain only coherent word pairs. The new registrations look the most malicious among malicious sets except JoeWein's list with its 16.5%. It tells us that newly registered domains tend to look like malicious domains, much more than domains being there for a while. It might not be used as evidence, but it strengthens our belief, that most new registrations are indeed of questionable intent.

There are several points where our results can be improved. The percent of domains receiving zero point while tagged by the Wiki Module (meaning they contain only incoherent word pairs) is still high and over 60 percent for all domain sets. This means we still have a to improve our set of word relation, which is no wonder, as we used less than 30% of English Wikipedia pages, and it is not even proved that Wikipedia is comprehensive for finding relations between words.

5.2 Random Seeker Module

While most of the domain names contain at least two words as seen in Section 5.1.6 some don't contain any words or contain just one. The Random Seeker Module will try to find random strings in domain name. Even a domain name containing multiple words can have a random string appended. The advantage of this module, that it can be used for all domains (having 0, 1 or more words in them).

5.2.1 Design of the Random Seeker Module

The idea behind this module is we look at words in ordinary text and collect how often a letter appears after another letter. We suspect that in randomly generated domain names we will find a lot of letter combinations that we won't see in regular words

The Design of the Random Seekers Module is very similar to the Wiki Module's (Section 5.1.1). It has a module responsible for generating bigrams from the Wikipedia articles called Letter Bigram Module. The biggest difference is that the Letter Bigram Tagger Modules directly uses the domains gained from the Domain Extractor Module (Figure 5-14).

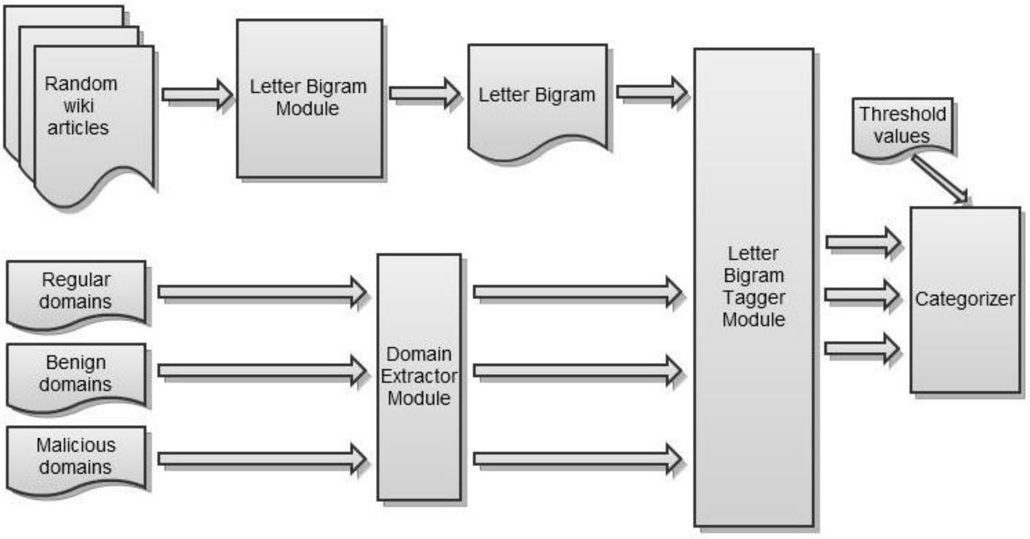


Figure 5-14, Design of the Random Seeker Module

The Letter Bigram Module will search for the paragraphs in the Wikipedia articles. It will break the paragraphs to sentences and the sentences to words. For each word, it will strictly find one-way relationships between letters. From the word spam it will find s-p, p-a and a-m relations but will not find the inverses (p-s, a-p and m-a).

The tagger module will point the domain names based on the letter pairs found in them. If letter pairs found in domain names are frequent, it will give 0 point for that pair. If a pair is infrequent, it will get a score. There are only 676 possible letter combinations. The last 10 percent will get the highest score, the next 10 percent will get a somewhat lower score and so on. But only letter combinations in the lower 50% will get points for a domain. Hyphens and numbers in a domain name will add to the score but only a few points. When we evaluate the total points for a domain we will divide it by the length of the domain name, because being long doesn't mean being random. Then finally, we can set different threshold values to tell if a domain is random.

5.2.2 Evaluation of Results

The letter bigram has been created for both the downloaded 200000 Wikipedia articles and for the 1.1 million set too. The result was interesting: the heat map for both sets looked exactly the same (Figure 5-15). It means that we have probably found a letter bigram, which is representative for common English text on the Internet (at least for Wikipedia).

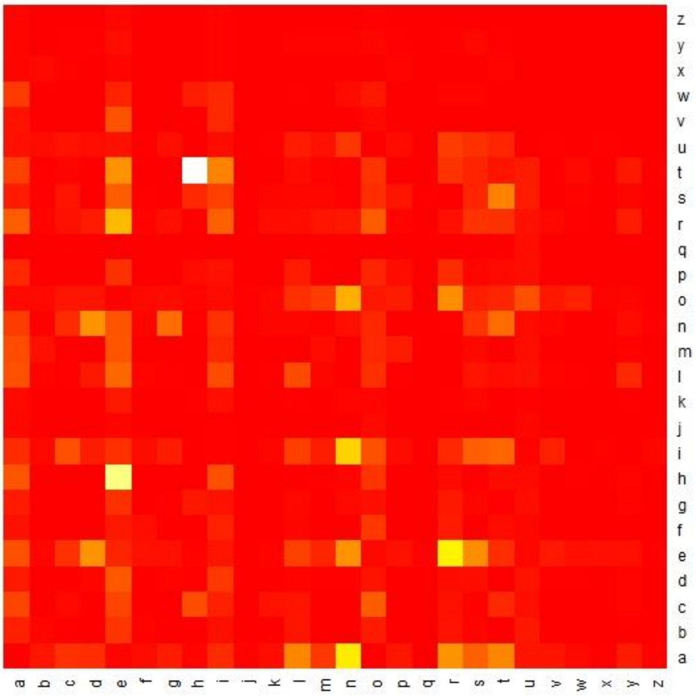


Figure 5-15, heat map of the letter bigram

In Figure 5-15 whiter colors mean high frequency, and red color means low frequency, just like in the heat of flame. The highest frequency is t-h, and the second highest is h-e. Some of the lowest in order are: q-j, q-z, q-g, j-q and z-x pairs.

To evaluate how random a domain name is, I used four categories based on the points given by the Letter Bigram Tagger Module. Domain names with zero or very low points will be in the not random category. There are categories: could be random, possibly random and probably random. Probably random domain names have a good chance of being random, accordingly I named it strong random, while the other two random categories were named weak random.

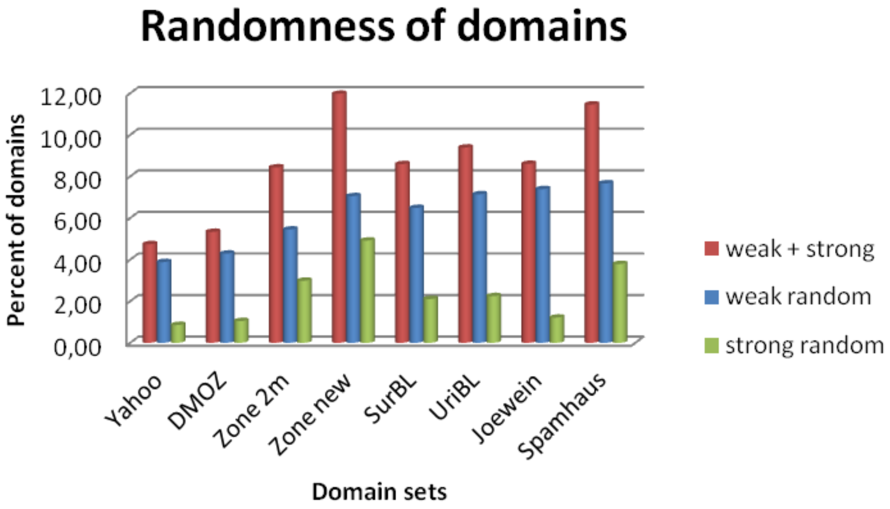


Figure 5-16, randomness of domains

In Figure 5-16 results show, that the benign sets have a low value around 5 percent random domain names in them. The malicious sets on the other hand have over 8 percent of random domain names, which is for JoeWein over 11%. Domains from the new and existing zone file dataset look like the malicious domain sets (just like in Section 5.1) and the new registrations coming out as winners with approximately 12% of domains being in one of the three random categories. This is nearly three times more than in the Yahoo set!

While looking at the results in a lot of cases I have seen random domain names actually being combinations of abbreviations, for example: scjrtn.com, which stands for St. Croix Jack Russell Terrier Network, a page about dogs. To gain a statistical knowledge of how many of the random domain names we found are really random I used manual sampling. For each of the four categories I collected 105 domains randomly altogether 420 domain names. For the weakest random set, I found only 20%

of them being random, for the possibly random set, I found 44.76% of them being random and for the probably random set, I found 74.28% of them to be random. False positive rate for the not random set was as low as zero, because I did not find any random domain names in this set.

Using the result of manual sampling I have corrected the results gained by the Random Seeker Module Figure 5-17 shows the sum of the corrected percent of the random domains.

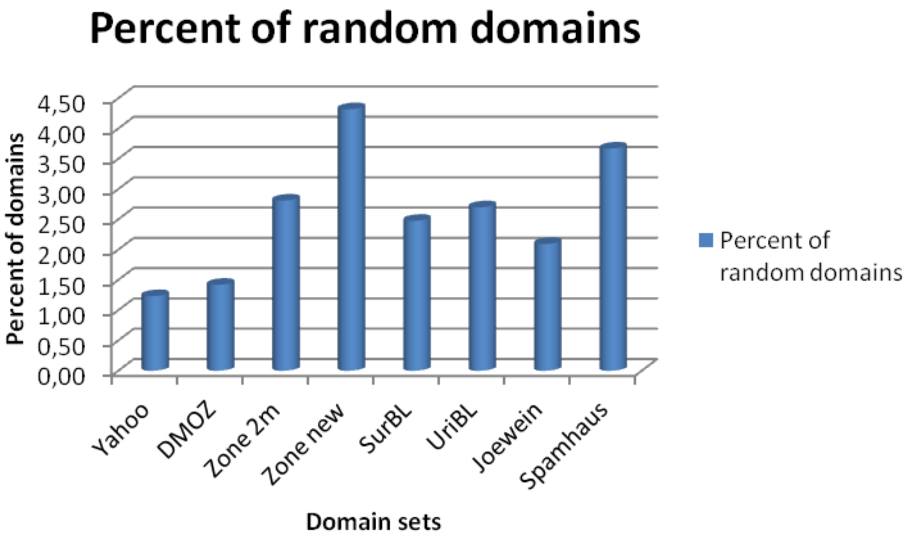


Figure 5-17, percent of random domains

We can see that the benign sets Yahoo and DMOZ only contain 1.23% and 1.42% random domain names respectively. The malicious sets contain much more random domains: 2.48%, 2.7%, 2.09% and 3.67% for Surbl, Uribl, JoeWein and SpamHaus. The regular sets, random Zone domains and new registrations look similar to the malicious sets (of 2.81% and 4.32% random domains). These values in some cases are three or nearly four times higher than the values of the benign sets.

This feature looks like it can differentiate between malicious sets of domains from benign sets, and shows that the regular sets tend to look like the malicious sets. Random Seeker Module could be used with some improvements as an effective tool to find random strings.

In the thesis, we focused on the development of the crawler and categorizer framework. Although we performed a number of measurements on the datasets, several improvements are left for future work. Looking at random domains found in benign sets we often find that those domains look random because they are a combination of abbreviations. To evaluate this we could do manual sampling for each set separately. I would expect the outcome would benign sets having nearly 0% random domains, while for the other sets the number of random domains would increase a bit. The tool could be improved by connecting it with the Best Overlap Module, and with using some simple machine learning techniques. Best Overlap Module could help us to find words and abbreviations in domain names, and give zero or very low point to those letter combinations. Machine learning techniques could help us setting the threshold values to find as many random domains as possible with a low false positive rate. This could make correction with manual sampling results unnecessary.

5.3 Conclusions and future improvements

During the analysis done in Section 5.1 we found that for all domain sets (benign, regular and malicious), the number of words in domain names is high. The reason for this could be that most miscreants mass registering domains try to generate benign looking domain names and they successfully do this. We found that the Wiki Module can be used for more than 85% of domains in all sets.

The final results show a very significant difference between the number of domains surely not algorithmically generated for Yahoo and DMOZ sets against the Surbl, JoeWein, SpamHaus and Uribl sets. In benign sets sometimes the ratio of not algorithmically generated domains was nearly twice as high as in the malicious sets. This makes the Wiki Module capable to statistically differentiate malicious domain sets from benign ones.

Still very high percent of domains were tagged as containing only incoherent word pairs. This is due to the still not comprehensive enough set of word pair relations. We only used less than 30% of all English Wikipedia articles, using all of them would mean marginally more coherent word pairs to be found. But even then, it would be useful to include other texts from different pages and Wikipedia articles in other languages. The other modification that could improve the results is to log all the best or nearly best overlaps found in a domain and use only the one with the highest point given by the Tagger Module.

In Section 5.2 we have seen that while benign sets contain very few random domains, the malicious sets can contain three times more. This feature differentiates between malicious and benign, but still could be improved to find malicious domains with a low false positive rate. To improve we have to connect it with the Best Overlap Module and use some machine learning techniques.

We found that in each set the number of domains not containing any words is very low, but the Random Seeker has only found less than 3% of domains to be random in the com zone file. . This contradicts the conclusion of the study discussed in Section 3.6. There is no easy way to find the malicious domains by the randomly generated domain names not including meaningful words as our paper shows that this only applies for three percent of all com domains.

The results strengthened our belief that most newly registered domains are malicious as most of the results show that the regular domain sets followed the same lexical properties, as the malicious domain sets, and the new registrations looked even more malicious than the random com domain set.

6 The BIZ Top-Level Domain: Ten Years Later

6.1 Motivation

The com zone became the most popular TLD among all domain names and it became crowded in a short time. For the average Internet users com was synonymous with the Web and soon it also became part of the typical branding procedure of companies. To give alternatives on com, in 2001 ICANN introduced the biz and info gTLDs. The success of this endeavor hinged on how users would perceive TLDs: whether users accepted the notion of TLDs as simply reflecting different potential homes for various Internet entities, or they viewed “dot com” as the sole TLD where they would expect to find prominent Web enterprises. Will users be confused of registrations by different parties of the same subdomain in different TLDs?

A decade has passed. ICANN is promoting now a different more open system for introducing new TLDs, one that would dramatically change the Domain Name System. Hence, this is the right time to evaluate if the introduction of the biz and info TLD fulfilled the expectations. The lessons learned could serve as useful information at the introduction of new TLDs.

To make a point, let me show some parts of the interview (Dickinson, 2011) made by Business Insider (BI) with Esther Dyson (ED) former chairman of ICANN:

***BI**: Why did you testify against ICANN?*

***ED**: I was a founding chairman from 1998-2000. I thought it was a good idea but I have since changed my mind. [Back then], com was not as strong and prevalent. We thought ICANN would do a good job at making sure scam artists didn't proliferate. That unfortunately hasn't happened.*

It offers a huge opportunity for people to buy domain names. In the meantime, people will try to make money. We will have to spend a lot of money on protecting Trademarks.

It will be used for bad things. Because so many current domain names are being used for bad things. The current registrars really don't police them that well. The users are supposed to put real names and addresses in the filings and they don't.

...

It's going to get worse rather than better. They keep saying there's a shortage of domain names. No, there's a shortage of trademarks. There aren't enough words that add meaning. Creating more words isn't going to reduce the confusion, it's going to increase it.

...

We worked in cooperation with UC San Diego and International Computer Science Institute to investigate how ICANN's plans on expanding the TLD system will affect the usage of domain names. We inspect the case of BIZ top-level domain, and how it is used ten years after it has been introduced to the public. My work in this study was: creating a web crawler, crawling for the content and classifying the content. The results of our joint work will be presented at the Passive Active Measurements (PAM 2012) conference (Halvorson, et al., 2012).

6.2 Data and Methodology

For this study, we relied on the data sources introduced in Section 4.2. We set a date June 27, 2011, the 10th anniversary of the launch of biz TLD by ICANN. We obtained biz and com zone files of this date using three sets of domains: all 2.1 million biz domains, their 2 million com namesakes and a random sample of 2 million com domains (same as the one used in the study discussed in Chapter 5). Whois information, DNS probing and web content were used for the classification of domains.

Zone files (discussed in Section 4.2.2) were used to obtain biz and com domains and to find domains registered in both gTLDs and name server information of the domains were also gathered, for the DNS crawler.

Whois records (discussed in Section 4.2.3) were used to find similarity between biz and the corresponding com registrant information. To do this similarity check, the Whois record had to be parsed. For this task a customized version of PHPWhois (SourceForge: PHPWhois) was used. We couldn't obtain all Whois information for two reasons. Reason one: many registrars limit the accessibility of their Whois database. Reason two: each of these registrars have different format for their Whois data, and PHPWhois couldn't parse all of them. Over-all, we were able to extract registrant information for only 65% of our biz-com pairs.

To assess similarity of a biz-com pair, we computed for both of them the Damerau-Levenshtein distance (described in Section 3.2) of the registrants' information: name, e-mail address, phone number, fax number. Fields were marked as missing when we couldn’t find them, and we also marked fields missing when they were less than five characters long. If a field was not missing and the Damerau-Levenshtein distance between biz and com was less than or equal two we marked it as a match. Also for registrant name, we marked it as a match if the com domain name was substring of the biz domain name, or vice-versa.

We defined strong match when at least two fields matched and maximum one field was missing. Weak match was termed when any of the four fields matched for the biz-com pair. Strong and weak match were differentiated because significant number of com domains only contained registrant name but no further fields, therefore we couldn’t find strong match for them.

Using the **DNS crawler** for each of the domain sets (2.1 million biz domains, their com pairs and for the 2 million random com domains) we queried for their name servers’ DNS records. If the name server answered our query, we queried it for three different names. For example if it was the name server of example.biz we would query for example.biz, www.example.biz and random.example.biz (where random was a ten to twenty character long, generated random string).

Web crawling was my part of the job. I created a multithreaded crawler, which was run on multiple machines (to speed up the process of downloading millions of pages), so at a time thousands of domains were downloaded with dozens of crawlers working. I collected the content for all domains in our dataset. First, I used the crawler to download pages for the domains in the biz zone, e.g. example.biz. Then, I crawled the corresponding com domain foo.com. Also, the contents of the 2 million random com domains were collected. When downloading the pages I recorded the HTTP Status codes: success, redirection and errors; and other standardized events: time out, DNS record not found, HTTP exceptions and other errors. Many of the Web pages were unreachable, either because the domain names were not intended to provide Web content, or due to the time difference between the zone file and our active Web crawling.

I created a simple **content classifier** (designed in the fashion described in Section 4.1) that searched for a set of regular expressions in the downloaded content to help us to identify parked domains. We created two classes of regular expressions: high-specific patterns and word-combinations. High-specific regular expressions were created from known parking site templates, relying on features of the page only characteristic for a given group of parking sites. Some word combinations are typical for many parked domains, e.g., buy domain, domain sale, and so on.

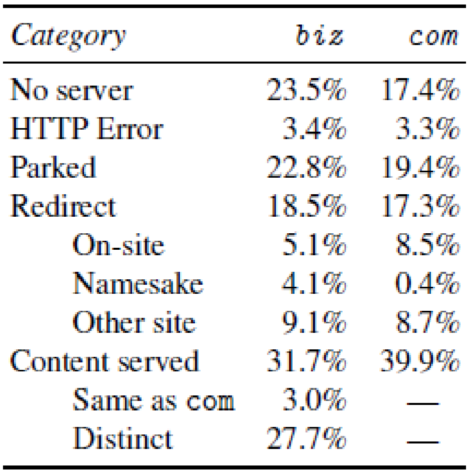
Five attributes were collected for each domain:

- **Registrant.** The name and address of the registrant and registrar from the Whois record
- **DNS server.** The name and IP address of the authoritative name servers for the domain
- **DNS records.** A and CNAME records for domain.tld and select subdomains
- **Web server.** HTTP status codes or redirects returned from domain.tld
- **Web content.** The web content obtained after following any HTTP-level redirects

6.3 Analysis

We wanted to evaluate how many of the biz domains are primary domains, defensively registered domains, or parked domains. Domains that are used to identify a company, product, service, or organization are **primary** domains. Primary domains are actually used by the registrant. The domain crysys.hu is a good example as it is used by the CrySyS Lab (the laboratory of the Budapest University of Technology and Economics) to identify itself on the Internet (or skateordie.hu is used by SkyRunner Kft.). Domain registrations are **defensive** (Section 2.2.1.6) if they are used to protect the name, but not used by the registrant to identify itself, its service, or network resources. A well-known example is Google having many defensive registrations for example google.biz and gooogle.com which both redirects too google.com. A **parked domain** is one, which is not in active use by the registrant, and which does not represent a name or brand used by the registrant (more in Section 2.2.1.5).

Figure 6-1 shows the result of content analysis described in Section 6.2.

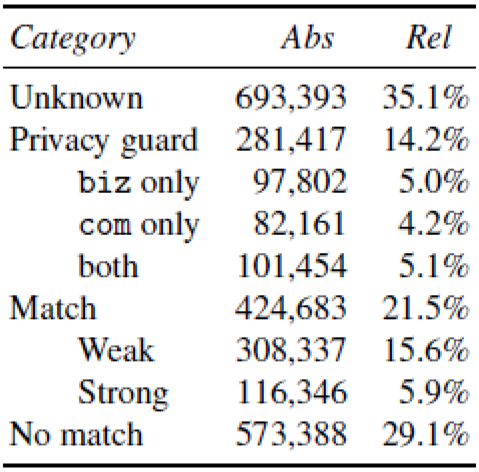


Category	biz	com
No server	23.5%	17.4%
HTTP Error	3.4%	3.3%
Parked	22.8%	19.4%
Redirect	18.5%	17.3%
On-site	5.1%	8.5%
Namesake	4.1%	0.4%
Other site	9.1%	8.7%
Content served	31.7%	39.9%
Same as com	3.0%	—
Distinct	27.7%	—

Figure 6-1, the Web behavior of domains in the biz and com gTLDs. The biz column shows statistics for the 2.1 million domains in the biz TLD, the com column for a random sample of 2 million com domains.

Confirming the registrant's intention on the usage of a domain is difficult, but some characteristics of the domain may strongly suggest defensive use. In our analysis, we considered four indicators of defensive registration.

In a biz-com pair if both domains registered by the **same registrant** it is perhaps the strongest indicator of defensive registration. We used the available Whois data (Section 6.2) to find domains registered by the same entity. From the successfully retrieved and parsed Whois records (65% of all biz-com pairs), we found 10.1% of biz domains and 9.3% of com domains had some manner of "privacy protection" mechanism, leaving out the registrant information from the Whois record. Overall, we had 50.6% of all biz-com pairs to evaluate by the common registrant feature. We determined that 5.9% of the pairs were strong match, 15.6% were weak match, and 29.1% were unlikely to be registered by the same registrant (see Figure 6-2).



Category	*Abs*	*Rel*
Unknown	693,393	35.1%
Privacy guard	281,417	14.2%
biz only	97,802	5.0%
com only	82,161	4.2%
both	101,454	5.1%
Match	424,683	21.5%
Weak	308,337	15.6%
Strong	116,346	5.9%
No match	573,388	29.1%

Figure 6-2, comparison of registrants of biz domains and their com namesakes using Whois records, showing absolute and relative number of biz–com name pairs in each category. Rightmost column shows value relative to total number of domain biz–com name pairs (93.8%).

Biz domain and its com namesake domain **sharing infrastructure** can be observed by DNS crawling. When there is a record same for the biz domain and its com counterpart, such as a CNAME pointing to the same domain or having the same A record it indicates a significant relationship between the biz and the com domain. 22% (452463) of biz domains were found to be sharing infrastructure with their com counterpart. The relation between the two domains can mean three things: it can be defensive registration, it can be just coincidental common hosting, when the owners of biz and com domains are unrelated, but they use the same hosting infrastructure, or it can be that both domains are parked using the same domain parking infrastructure.

Common CNAMEs, which would show a clear relation between two domains, are found in only 1.5% of all domains. In case of common A records, we did not have a way to distinguish between the first two cases, but we expected that coincidental Web hosting is less likely. We can distinguish domain parking cases by excluding from our set of domains the ones found to be parked by content classification. If we exclude all domains we categorized as parked we still find commonality in 280408 out of 1589203 (18%) biz-com pairs. This suggests significant fraction of biz domains to be defensive registrations.

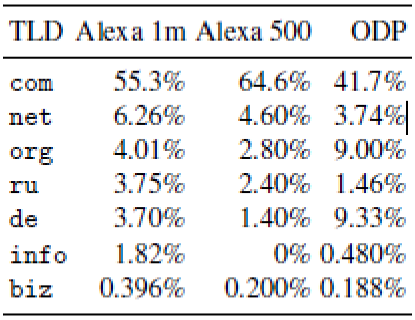
Entities registering domains defensively often redirect all traffic to the primary domain. In many cases, **HTTP redirection** is used, as it is visible to the browser and has the advantage of changing the user-visible address bar to reflect the new address (showing the primary domain, consistent with the branding of the site). Figure 6-1 shows that 18.5% of biz and 17.3% of com domains host a Web server that redirects the user. 4.1% of biz domains redirects to the namesake in a different TLD, while it is 0.4% for com domains. It means biz is nine times more likely to redirect the user to a namesake in a different TLD than com.

Redirection is not the only mechanism to have the same content appear on the defensively registered domain as on the primary domain. Simply maintaining the same page with identical content is used as a defensive mechanism. We found that 3% of non-parked sites did indeed serve the same content.

We hand classified 259 domains chosen at random from the 2 million domains appearing in both biz and com zone. We were unable to confidently classify 15%, the remaining were 35% parked, 25% primary, 26% defensively-registered

6.4 Discussion and conclusions

To get a feeling of the fundamental value of biz TLD to registrant, we should answer the question how many biz domains are being actively used by their registrants. Figure 6-3 shows that in Alexa's top 1 million list com domains appear 140 times more frequently than biz domains, the top 500 com appear 323 times more frequently than biz and com is 218 times more frequent than biz in the DMOZ directory. Com zone is only 46 times larger than biz, but proportionally biz is even much less popular than this would indicate.



TLD	Alexa 1m	Alexa 500	ODP
com	55.3%	64.6%	41.7%
net	6.26%	4.60%	3.74%
org	4.01%	2.80%	9.00%
ru	3.75%	2.40%	1.46%
de	3.70%	1.40%	9.33%
info	1.82%	0%	0.480%
biz	0.396%	0.200%	0.188%

Figure 6-3, TLD frequency in the Alexa listings and the Open Directory Project. In the Alexa 1,000,000, biz ranks (in frequency of occurrence) between com.cn and ir, while in the ODP, it falls between cat and za. Only one biz domain, livedoor.biz (a blogging site)

We found 22.8% of domains to be parked (by a known parking service). We could certainly tell for 10.4% of domains that they are defensive registrations. 27.7% served some content (excluding cases where it was identical to the namesake).

We found that in many ways biz resembling com domains, most remarkably, in parking biz is like com. In many other ways, biz couldn’t compete with com. It did coerce some existing domain owners to register domains defensively. While registering a domain costs just $10 it is an additional cost of defending trademarks.

On the eve of a bold new initiative by ICANN to open TLD registration to the general public, the history of biz provides a valuable lesson in costs and benefits related to DNS name space expansion.

7 Summary

I have learned a lot while writing the thesis. I have gained a better understanding of the DNS infrastructure and history. I have also deepened my knowledge on the speculative and malicious usages of domains. Studying related work, I could clearly see which sources are available for domain classification. I have seen quite a few methodologies and features that can be used to take advantage of the information available for a domain.

I designed and used a framework to ease the effort of evaluating features and it has made my job much easier. During the research done in this thesis, we explored many features. Testing the first features did not meet our expectations, neither DMOZ nor OpenDNS proved to be a useful tool for direct topic wise categorization of domains; they could only be employed for extracting attributes typical of a topic. Using heuristic methods, we were able to evaluate that some features are not capable of categorizing most of the domains (features based on hyphens or numbers in the domain name). It helped me though to a better and deeper understanding of how existing domains look and built, and helped me to select features more fitting for the thesis.

We could see in many cases that the words in malicious domains are not related. Building upon this idea, I have created a module, which uses English Wikipedia articles to evaluate the coherence of words in a domain name. While the first version of the Wiki Module couldn’t differentiate between malicious and benign domain sets, after making improvements it started to show some interesting results. Using a multithread crawler I designed, I have downloaded over 1.1 million random English Wikipedia articles. I selected the most frequent words (about 0.6 million) from the nearly 1.9 million word found in the downloaded Wikipedia pages and extended it with a UK dictionary to find best overlaps in domain names and to evaluate the number of words in domain names. I have found for all domain sets (benign, regular and malicious), that the number of words in domain names is very high, indicating that Wiki Module can be used for more than 85% of domains in all sets. This confirmed our idea that malicious domains successfully resemble to benign domains as much as possible.

I have extracted over 114 million word pair relations to use for tagging purposes from the 1.1 million Wikipedia pages. The results show that for some benign sets the ratio of not algorithmically generated domain names was nearly twice as high as in the malicious sets. Therefore, the Wiki Module can be used to differentiate malicious sets from benign sets.

I looked at randomly generated domain names and discovered that their number is very low (less than 3%) for com domains. This fact warns us that most of the domains can’t be analyzed this way. However, Random Seeker Module can be used after improvements to categorize randomly generated domains with a low false positive rate.

All research shows that the regular domain sets follow the same trends, as the malicious domain sets, and the new registrations look even more malicious than the random com domain set. This result strengthened our belief that most of the newly registered domains are malicious, indeed.

We have addressed several questions of domain registration, but there is ample room for improvements in future work. The Wiki module can be improved in many ways. We still don’t have comprehensive enough set of word pair relations: extracting relations from more pages could be a way to improve. Also using more best overlaps when coherence of word pairs is evaluated in a domain, could eliminate those cases when the best overlap module chooses the wrong set from two equally long best overlaps. Quality is an important factor that should be improved for the word list we use and with adding some grammar rules to the Best Overlap Module.

The Random Seeker Module can be improved by connecting it with the Best Overlap Module to manage words apart from random string parts, and with using machine-learning technique to help us set the threshold values.

Studying how biz is used ten years after it has been introduced to the public, we see that it has caused many defensive registrations, and that the biz zone contains a lot of parked domains. We found that biz is disproportionally less popular than com domain. When corporations are suing ICANN for the new gTLD xxx and when ICANN plans to introduce a new, more open system of TLDs, the history of biz provides a valuable lesson in costs and benefits related to DNS name space expansion.

Acknowledgements

I would like to express my gratitude to all those people, family members, friends, associates, colleagues and lecturers who gave me their support to complete this thesis.

It was good to work with the domain doctors on the article “The BIZ Top-level Domain: Ten years later” for Passive and Active Measurement conference (*PAM 2012*). I am grateful for the experience and knowledge gained from the cooperation. Thank you, guys.

I am indebted to my supervisor, Mark, whose assistance, inspiring suggestions, support and encouragement helped me all the way during research and writing of this thesis.

I needed high computational power for my research, much more than my laptop was able to provide. My work could not have been completed as it is, if my brother and his associates did not give access to their servers having enough capacity to make the complex research necessary for this thesis. I also want to thank Tozo for configuring the servers for my use and allocating resources for me.

I am grateful to my father for reviewing my English and grammar in this paper, and taking the responsibility for the mistakes left in the final version.

I would like to give my special thanks to my mother whose patient love enabled me to complete this work, and to my sister Andrea for being my best partner in recreation. Last but not least, I want to express gratitude to my girlfriend Orsi, for supporting me throughout the difficult period of writing this thesis.

Bibliography

Alexa: Alexa top sites. (2011). Retrieved 2011, from Alexa: http://www.alexa.com/topsites

Bilge, L., Kirda, E., Kruegel, C., & Balduzzi, M. (2011). Exposure: Finding malicious domains using passive dns analysis. *Proceedings of NDSS.*

celeranetworks.com: Spam Solutions. Retrieved December 16, 2011, from celeranetworks.com: http://www.celeranetworks.com/solutions/spam-solutions/

Coull, S. E., White, A. M., Yen, T.-F., Monrose, F., & Reiter, M. K. (2010). Understanding Domain Registration Abuses. *Security and Privacy–Silver Linings in the Cloud* , 68-79.

Dickinson, B. (2011, December 13). *Tech: Business Insider*. Retrieved December 16, 2011, from Business Insider: http://www.businessinsider.com/boonsri-dickinson-esther-dyson-tried-to-buy-an-xxx-to-protect-meetupcoms-brand-2011-12

DMOZ: ODP – Open Directory Project. (2011). Retrieved 2011, from DMOZ: http://www.dmoz.org

Faloutsos, M., Markopoulou, A., & Le, A. (2011). PhishDef: URL Names Say It All. *INFOCOM, 2011 Proceedings IEEE* (pp. 191-195). IEEE.

Felegyhazi, M., Kreibich, C., & Paxson, V. (2010). On the potential of proactive domain blacklisting. *Proceedings of the 3rd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more* (pp. 1-6). San Jose, California: USENIX Association.

Halvorson, T., Szurdi, J., Maier, G., Felegyhazi, M., Kreibich, C., Weaver, N., et al. (2012). The BIZ Top-Level Domain: Ten Years Later. *Passive and Active Measurement Conference 2012,* (pp. 1-10).

He, Y., Zhong, Z., Krasser, S., & Tang, Y. (2010). Mining DNS for Malicious Domain Registrations. *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010 6th International Conference on* (pp. 1-6). IEEE.

HosterStats.com. Retrieved December 16, 2011, from http://www.hosterstats.com/DomainNameCounts2011.php

JoeWein. (2011). Retrieved December 16, 2011, from JoeWein: http://www.joewein.net/

Kurose, J. F., & Ross, K. W. (2009). *Computer Networking.* Pearson/Addison Wesley.

M86security.com: Security Labs Report, January -June 2010 Recap. (2010, June). Retrieved December 16, 2011, from M86security.com: http://www.m86security.com/documents/pdfs/security_labs/m86_security_labs_report_1H2010.pdf

MAAWG: Email Metrics Program. (2011, November). Retrieved December 16, 2011, from MAAWG: http://www.maawg.org/sites/maawg/files/news/MAAWG_2011_Q1Q2Q3_Metrics_Report_15.pdf

Moore, T., & Edelman, B. (2010). Measuring the perpetrators and funders of typosquatting. *Financial Cryptography and Data Security* , 175-191.

Ollmann, G. (2009). Botnet communication topologies.

OpenDNS: Domain Tagging. (2011). Retrieved December 16, 2011, from OpenDNS: http://community.opendns.com/domaintagging/

Pepitone, J. (2011, June 20). *CNNMoneyTech*. Retrieved December 16, 2011, from CNN: http://money.cnn.com/2011/06/20/technology/dot_brand_domain_name_icann/index.htm

PHPWhois. Retrieved from http://sourceforge.net/projects/phpwhois/

Rader, R. W. (2001). One History of DNS.

SourceForge: PHPWhois. Retrieved from SourceForge: http://sourceforge.net/projects/phpwhois/

Spamhaus: DB. (2011). Retrieved December 16, 2011, from Spamhaus: http://www.spamhaus.org/dbl/

Spring, J. M., Metcalf, L. B., & Stoner, E. (2011). Correlating Domain Registrations and DNS First Activity in General and for Malware. *Proc. Securing and Trusting Internet Names (SATIN)* .

Surbl: lists. (2011). Retrieved December 16, 2011, from Surbl: http://www.surbl.org/lists

UriBL. (2011). Retrieved December 16, 2011, from UriBL: http://www.uribl.com/

Wikipedia: Cybersquatting. (2011, December 6). Retrieved December 16, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Cybersquatting

Wikipedia: Domain name. (2011, December 7). Retrieved December 16, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Domain_name

Wikipedia: Domain name front running. (2011, November 10). Retrieved December 16, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Domain_name_front_running

Wikipedia: Domain Name System. (2011, December 13). Retrieved December 16, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Domain_Name_System

Wikipedia: Domain parking. (2011, October 19). Retrieved December 16, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Domain_parking

Wikipedia: Generic top-level domain. (2011, December 14). Retrieved December 16, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Generic_top-level_domain

Wikipedia: Phishing. (2011, December 14). Retrieved December 16, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Phishing

Wikipedia: random. (2011). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Special:Random

Wikipedia: Root name server. (2011, December 8). Retrieved December 16, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Root_name_server

Wikipedia: Spam (electronic). (2011, December 12). Retrieved December 16, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Spam_%28electronic%29

Wikipedia: Spam (food). (2011, December 15). Retrieved December 16, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Spam_%28food%29

Wikipedia: Typosquatting. (2011, November 20). Retrieved December 16, 2011, from Wikipedia: http://en.wikipedia.org/wiki/Typosquatting

Yahoo: random. (2011). Retrieved from Yahoo: http://random.yahoo.com/bin/ryl