Long-term Active Measurement and Evaluation of E-mail System Related DNS Entries

eingereicht von

Gabriel Alexandru Kovacs

Matrikelnummer 0425613

Bachelorarbeit

Studienrichtung: Medieninformatik
Fakultät für Informatik der Universität Wien

Betreuer:

Univ.-Prof. Dipl.-Math. Dr., M.A. St. Peter Reichl

Abstract. E-mail has been the first Internet application which had a considerable impact on how people interact in today's world making it an indispensable technology. At the time is was developed no security measures were built in thus leaving the door open for attackers and making those who use e-mail vulnerable. To counter this issues DNS based solutions were developed which try to authenticate the sending domains thus reducing the risks. This work presents some of these solutions (SPF, DMARC, DKIM, ADSP) and there deployment rate by measuring the DNS entries of the domains available in the Alexa Top 1 million list for a period of 10 months (between 2014 and 2015). After providing the findings we present the next steps which we would like to undertake.

Contents

1 Introduction

1.1 Motivation

Internet has been growing tremendously in the past decades and its strategic importance is comparable to that of modern road systems. As means of communications e-mail has established itself since the beginnings of the Internet and still prevails. E-mail has become a cheaper and faster replacement of the classical postal service, having a considerable impact on productivity. Even with the rise of social media and the breakthrough of various mobile devices e-mail remains the backbone of today's conversational system. Because of its commercial importance steps should be taken to consolidate and improve e-mail security thus reassuring the general public, that there are no risks regarding e-mail use (that no harm can come from using e-mail).

1.2 Problem Statement

At the time e-mail was developed no measures regarding security were necessary and have thus not been taken into consideration. Beginning with the proliferation of the Internet in the 1990ies the need for e-mail security become apparent for companies and the general public due to the appearance of spam and phishing. Spam is an unwanted e-mail mostly used for advertising but also for fooling people in order to steal their money, by for example fraudulently promising a high gain for a low investment or some extraordinary discounts for product purchases. Usually spam e-mail can be easily recognized be experienced users due to the high number of misspelled words and sometimes nonsense content present in the body and header. Phishing is much more sophisticated then spam e-mail due to its setup and purpose. In comparison to a spam e-mail a phishing e-mail will always try to conceal its origin and furthermore imitate an organization (bank, e-mail provider) the user trusts. Phishing e-mails are designed to look like official e-mails sent by a trusted organization asking the user for confidential information or providing a link to a website that emulates the original website prompting the user to login in to his account. A successful phishing attack can have dire consequences for the targeted person as well as for the impersonated organization. The victims of a phishing attack can have their bank accounts plundered, social media (facebook, twitter) and e-mail accounts hijacked, get their computers infected with malware which can lead to theft or loss of personal data. The impersonated domain will more than likely lose its reputation which in case of financial institutions and e-commerce companies can lead to a diminishing customer base.

Not only individuals may be targeted by phishing attacks but also companies or organizations. A notable recent target of a phishing attack was ICANN[1] (The Internet Corporation for Assigned Names and Numbers) in late November 2014. The attack was partially successful, thus compromising the credentials of several ICANN employees and their business partners. Additionally the attackers gained access to all the TLD Zone Files[2] in their system. This clearly demonstrates the severity of potential impacts especially when targeting organizations or persons of specific interest.

The internet community has long been struggling to confine and minimize the risks which arise by using e-mail, having partially succeeded but never being able to totally eliminate them. The next section will present different measures that are already in use and have been established and additionally some methods which are up and coming.

1.3 Solution

Because there is no built-in security in Simple Mail Transport Protocol (SMTP) other solutions had to be found to compensate this shortfall. Some of which take effect during the SMTP exchange or right after the SMTP transaction and the delivery of the e-mail to the users inbox. However until today, no holistic solution addressing all security deficiencies has not yet emerged.

In the early phase simplistic concepts such as Black Lists (BL), White Lists (WL) and Grey-listing (GL) have been deployed. BLs block all IP addresses of known spam-sending or otherwise malicious domains (e.g., cousin domains) based on list that is collected at the mail server or retrieved from external sources. Likewise, WLs are a list of IP addresses, which can always be trusted thus any e-mail being delivered from an IP address contained in the list can be delivered without further inquiries. Grey-listing is a technique intended to fool those bots which are responsible for sending out spam. The taken approach is as follows: when a suspicious e-mail arrives the receiving server will notify the sending server that the sent e-mail cannot be processed currently. If the sending server attempts to deliver the same e-mail again it will be accepted. This technique works because most of the bots sending spam do not attempt to resend the e-mail. Despite their benefits, BL and GL techniques have some drawbacks: Once one's IP address is a part of a BL it is very difficult to get it unlisted, which is further multiplied by inter-organizational exchanges of

[1] `https://www.icann.org/news/announcement-2-2014-12-16-en`. Retrieved: 2015-03-12.

[2] TLD Zone Files: contain domain names, their corresponding name servers and the IP addresses which identify the name servers.

BLs. Although e-mail was not designed to be delivered instantaneously, due to the technological progress it is today perceived and expected to be almost instantly available. By using GL the delivery of the e-mail will be delayed for a defined period of time (typically ranging between five minutes and one day), thus lowering the benefits of e-mail conversations. In some cases legitimate as well as spam and phishing e-mails are sent from the same IP address which make BL rather obsolete. A permanent GL approach is also not desirable considering the high load it can cause.

A more elaborate approach to combat spam and phishing are spam filters which are able to verify incoming as well as outgoing e-mails. The most popular spam filter is SpamAssassin due to its large community support, flexibility, ease of configuration and its free of charge principle. SpamAssassin is based on a set of heuristic rules which can be extended by e-mail administrators and it also offers the possibility to make use of third party libraries or programs like antiviruses. One of the features that made e-mail so popular was the possibility to attach data (e.g., different files) to the e-mail to be sent. Unfortunately attackers spread via this attachments malicious content which can damage or compromise one's personal information. To prevent this from happening antivirus software is used which can determine if an e-mail contains malicious content or not. Spam filters are very effective at what they do however there is a significant downside namely their usage is very resource-intensive. For this reason spam filters are used as a last resort in separating spam and phishing attacks from legitimate e-mail exchanges.

This led to a call for solutions that perform at least as good as spam filters but at the same time are able to be more resource efficient. This work presents some of the techniques that were developed, which try to meet today's challenges regarding e-mail security. The next section will describe the way they are deployed and there functionality.

1.4 Structure of the Thesis

The rest of the work is structured as follows: Section 2 provides the fundamentals of Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM), Author Domain Signing Practices (ADSP) and Domain-based Message Authentication, Reporting and Conformance (DMARC) and proceeds in Section 3 with the related work. In Section 4 the Measurement Setup is described while Section 5 specifies how the acquired raw data is further filtered and processed. Section 6 presents the adoption rates and trends of the e-mail security technologies based on the data gathered during the 10 months measurement campaign and finishes with a summary (Section 7) of the work.

2 Fundamentals

SMTP does not provide means of authenticating the e-mail sender which lead to a security loophole. To close this loophole path and signing-based technologies have been developed. The examined technologies are SPF [1] , DKIM [2] , ADSP [3] and DMARC [4]. All four rely on DNS entries, defined in the domains own name server (NS), in the form of TXT Resource Records (RR).

SPF is a path-based mechanism which verifies if the domain, which appears in the return path, has authorized the sending server to send e-mails on its behalf. This is achieved by publishing a TXT RR which contains the IP addresses of all legitimized e-mail servers and also the policy which the domain owner wishes to enforce on the receiver side. This policies range from “accept every e-mail” to “reject every e-mail” which does not conform. SPF was designed to prevent others from spoofing one’s identity. The weak point of SPF is forwarding and relaying which can break SPFs authorization process and can lead to legitimate e-mail being dropped. To prevent this from happening mechanisms like SRS (Sender Rewriting Scheme) have been proposed.

DKIM is a signature-based mechanism which allows the sending domains to assume responsibility for the e-mails they send. On the sending side DKIM encrypts parts of the e-mail and "attaches" this signature to the e-mail to be sent. On the receiving side the signature can be decrypted and verified with the help of a public key which can be found in the originating domains DNS in the form of a TXT RR. DKIM is a reputation based system which does not enable the sender to define a policy which should be taken in consideration by the receiver. As a consequences if the DKIM signature cannot be verified, it is up to the receiver to decide how to handle the incoming e-mail. The main reason for a DKIM fail is due to a modification of the e-mail body and/or header which usually occurs when the e-mail address is subscribed to a mailing list.

ADSP was developed as a complementary technology on top of DKIM to allow the sender to define a policy regarding its signing practices. The sender defines in its own DNS a TXT RR which allows him to specify three distinct policies:

- “unknown”: The signing practices are not precisely defined.
- “all”: All outgoing e-mails are signed.
- “discardable”: All outgoing e-mails are signed and furthermore if the signature cannot by verified the e-mails should not be delivered.

ADSP was not widely deployed and was said to generate more harm than good, thus the Internet Standard was downgraded to a Historic state.

DMARC can be considered an improved successor of ADSP which builds upon ADSPs principles but also extends its functionality. Just as ADSP, DMARC also defines three similar policies “none” (= unknown), “quarantine” (= all), “reject” (= discardable) and it also relays on a DNS entry in the form of a TXT RR. DMARC makes use of DKIM and SPF thus having a fallback in case one of them should fail or a second confirmation which legitimizes the incoming e-mail. Another feature is the possibility to define a policy not just for the main domain but also for all subsequent subdomains which can be implicit or explicit defined. The most important innovation is the possibility to request a report which provides details regarding the passes/fails of DKIM/SPF, if the e-mail got delivered or not and the IP address of the alleged spoofers.

All of the above technologies use the existing DNS infrastructure and at the same time they can be implemented with no or minor effort. They can be very efficient against phishing and are also able to combat spam if the sender has been spoofed. One of the weaknesses is the DNS system itself. If the NSs of a domain are unreachable or misconfigured no statement can be made regarding SPF, DKIM or DMARC. Another problem arises due to badly configured records which are the result of misunderstood standards and/or guidelines.

3 Related Work

This section presents the directly related works. We will in particular revise some of the previous measurements, performed by various organizations regarding the deployment of SPF, DKIM and DMARC.

The website *spf-all.com*[3] presents figures featuring the deployment of SPF, offering detailed information about the used policies. It uses domain lists from the years 1997, 2003 and other resources which are not verifiable. It is unclear if the initial domain lists were ever updated, when the measurements were made and also information regarding the measurement setup is missing.

BuildWith[4] offers SPF, DKIM and DMARC usage statistics. The measurements are based on the Quantcast[5] Top Million list and domains which have been gathered and merged in distinct domain lists by the company itself. They also offer charts which depict the deployment of SPF, DMARC and DKIM since the beginning of 2014 until February, 2015 based on a Top 1 Million list. According to the DMARC chart in March, 2014 none of the 1 million domains used DMARC. A list containing all the domains

[3] `http://spf-all.com/`. Retrieved: 2015-03-12.

[4] `http://trends.builtwith.com/mx`. Retrieved: 2015-03-12.

[5] `https://www.quantcast.com/`. Retrieved: 2015-03-12.

which use SPF, DKIM and DMARC can be purchased from the company's website.

Eggert[6] has been following the deployment of SPF and DKIM since October, 2007 and DMARC since August, 2014. Their measurements are ongoing and based on the Top 500 Alexa list for several Country-TLDs and the most popular domains which are updated before every new measurement. The number of the queried domains is rather small and also partially redundant due to websites which are popular all around the world. No other details are being provided.

The *Wide*[7] project tracked between April, 2005 and May, 2012 the deployment of SPF and DKIM for the Japanese Top-Level Domains in collaboration with the Japan Registry Service. They partially describe the used methodology and also provide some of their results.

Unlock the Inbox[8] implemented a feedback mechanism which can offer detailed information regarding one's own e-mail authentication system. The user has only to send an e-mail to a specific e-mail address and after a few minutes a detailed report will be send back. By providing this free service the company was able to gather information regarding the deployment of SPF, DMARC and DKIM. This method is the only one which can provide genuine data regarding the adoption on DKIM. They have been providing monthly figures since September, 2013 regarding the deployment of SPF, DMARC and DKIM. It is unclear how the data gets aggregated and how many distinct users make use of their service.

The above presented work is either outdated, lacks transparency regarding the methodology and in some cases provides questionable results. The work provided in this thesis closes the gaps by providing a detailed description of the measurement setup and data evaluation which has been collected through an extensive inquiry of the freely available data in the global DNS.

4 Measurement Setup

Figure 1 gives an overview of the steps which are necessary to perform the measurements and how the obtained data is further processed. The measurement setup shows the required actions which are necessary to acquire the desired resource records. The Python script reads in the Alexa domain list (point 0) and launches a series of DNS queries which are passed down (point 1) to the recursive resolver (Unbound). The resolver then makes use of the global DNS system by trying to resolve (point 2, 3, 4, 5) the domain

[6] `https://eggert.org/`. Retrieved: 2015-03-12.

[7] `http://member.wide.ad.jp/wg/antispam/stats/`. Retrieved: 2015-03-12.

[8] `https://www.unlocktheinbox.com/email-statistics/`. Retrieved: 2015-03-12.

name (example.com) and finally requesting (point 6) the resource records from the domain name server (NS). As soon as it receives a final response (point 5 or 7) from the name server the resolver will return the answer (point 6' or 8) to the Python script which stores it in a list. After the complete Alexa list has been processed the information contained in the list is written (point 9) to a CSV file which ends the measurement setup. The data evaluation begins with the first filter (Python script) reading in (point 10) the end results of every measurement and transforming the relevant data in a way which makes it more suitable for further processing. The thus prepared data passes through (point 11) a second filter (R script) which serves as input data for the tables and figures presented in this work.

In this section the focus will be on the measurement setup. The key feature of the measurement setup is the python script which initiates the DNS queries.

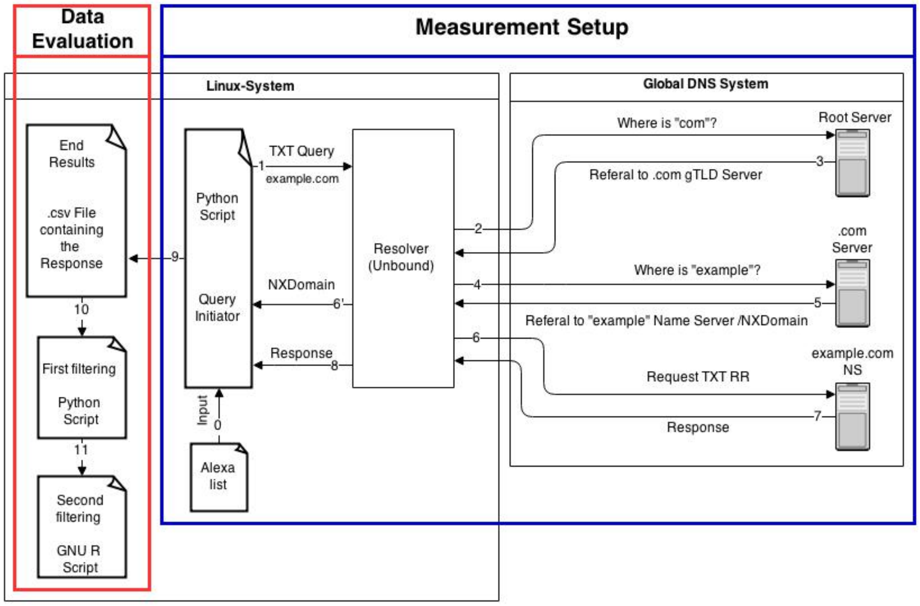

Fig. 1. Measurement and evaluation diagram.

As a production system a computer running Linux has been used with Python 3.3 as the programming language of choice. The first step was to determine how to obtain the resource records which every domain defines in their own DNS. For this task the python DNS module[9] has been used due to its longevity and the fact that is was still maintained.

Listing 1 shows a snapshot of the *get_ records* function which is responsible for initiating the DNS queries thus being called by all other functions which try to acquire a resource record. In line 2 a new resolver is being instantiated while in lines 3 and 4 the lifetime and timeout attributes of the resolver are being set. In line 8 the query is being performed and its output is being saved to the query list which gets returned by the function in line 9. The function also handles the DNS Errors which may occur during the DNS query process. **NXDomain** (Name Error) is an error message which occurs when the queried main domain is missing from the TLD Zone File. Such an example is showed in Figure 1 when querying for the TXT record of the example.com domain. At point 4 the resolver asks the .com server for the address of the name servers belonging to the example domain. If example is not listed in the .com Zone File a NXDomain message is being retuned (point 5) to the Resolver (Unbound) which further delegates (point 6′) the message to the python script. **NoAnswer** (No Data) is returned if the domain has not defined in its own DNS any of the requested Resource Records. **Timeout** is returned if no answer was provided in the predefined amount of time. This can occur due to a network congestion or the server being too busy. **NoNameservers** (ServerFail) is an error message which occurs when the queried name servers are misconfigured or unreachable.

The *resolver.timeout* which raises a Timeout error if the value is exceeded, was set to 120 seconds in accordance to [5], which states that after 120 seconds without response from a name server it can be assumed that the name server is unreachable. The *resolver.lifetime* has been set to 600 seconds resulting in a number of 5 attempts per queried resource record. Five was chosen because it corresponded to the maximum number of name servers observed during the previous measurements. The total time of 10 minutes was chosen so we could overcome potential network congestions and overloads of poorly configured name servers. For every domain the Python script tries to retrieve the following six resource records:

- The SPF record of the main domain.
- The A records of the main domain.
- The A records of the "www" subdomain.
- The MX records of the main domain.
- All TXT records witch can be found under the "_.dmarc" subdomain.
- All TXT records witch can be found under the "_adsp._domainkey" subdomain.

[9] `http://www.dnspython.org/`. Retrieved: 2015-03-12.

```python
def get_records(domain,recordType ,total_sec_per_attempts,
    sec_per_attempt):
  resolver = dns.resolver.Resolver()
  resolver.lifetime= total_sec_per_attempts
  resolver.timeout= sec_per_attempt

  try:
    query = []
    query = resolver.query(domain, recordType)
    return query

  except dns.resolver.NXDOMAIN:
    query.append("NXDomain")
    return query
  except dns.resolver.NoAnswer:
    query.append("NoAnswer")
    return query
  except dns.exception.Timeout:
    query.append("Timeout")
    return query
  except dns.resolver.NoNameservers:
    query.append("NoNameservers")
    return query
```

Listing 1. Get resource records function

To be able to perform multiple measurements per day Pythons multiprocessing module[10] has been used. This module grants the use of a pool of sub-processes which can distribute the work among all its workers thus leading to a much faster completion of the given task. Due to the task not being I/O-bound but rather CPU-bound a number of roughly 1000 workers (sub-processes) were spawned by the Pool which allowed the work to be finished in roughly 120 minutes. The pool returns a Python list which contains all the query results which afterwards are being written to a CSV file (point 9, Figure 1).

As input for the measurements the Alexa[11] Top 1 million list (dating from April 17th, 2014) has been used because at that time no other resources were available. From the list 14619 entries were removed because they did not provide any useful information. The deleted items were either IPv4 addresses or redundant and overly specific resource records (e.g., `youtube.com/user/EVOTV`) belonging to the same domain, thus always returning a NXDomain error message. The processed Alexa list serves as input (point 0, Figure 1) for the python scripts which initiates the DNS queries.

The used computer has been connected to the network of the University of Vienna, thus the queries are performed by the University's own resolver. To avoid answers cached by the resolver and also to prevent the resolver from overloading, it was necessary to use a dedicated resolver for this experimental measurement. The most known resolver is BIND, nevertheless due to the seemingly complicated setup a more configuration-friendly and

[10] `https://docs.python.org/3.3/library/multiprocessing.html`. Retrieved: 2015-03-12.

[11] `https://www.alexa.com/`. Retrieved: 2015-03-12.

lightweight solution has been used, i.e., Unbound[12]. Unbound does not require any additional configuration and can be used right after it has been installed. However to increase Unbounds performance threading (4 threads) was activated and for more security the Domain Name System Security Extensions[13] (DNSSEC) was also enabled.

Before starting our large-scale measurements on April 17th, 2014, a bash script has been written with the purpose of automating the measurement process. Inside an infinite loop before the beginning of every measurement Unbound is being restarted thus deleting its cache. After the measurement has ended the newly generated CSV file, a data file readable by most spreadsheet applications, is backed up on another machine. This led to a number of roughly 11 complete measurements per day.

5 Evaluation Methodology

This section will handle the data evaluation as seen in Figure 1. The focus will be on the first filter (point 10, Figure 1) which takes the end results acquired during the measurement and creates a new CSV file which will serve as input data for the last filter (point 11, Figure 1).

SPF, DMARC and ADSP records are defined with the help of a simple "tag = value" syntax where the separator could also be another sign not just the equal sign. An SPF record for example could look like this: "v=spf1 ip4:64.233.187.27 -all" where "v" represents the version tag, "=" is the separator and "spf1" is the value of the tag; similarly "ip4" represents the tag, ":" is the separator and "64.233.187.27" describes the value of the tag; "-all" describes the policy the sender wishes to enforce on the receiving side. Likewise there is a separator between the "tag=value" combination, in the case of SPF its white space. The SPF record would be understood as: Accept only e-mails which are delivered from the 64.233.187.27 IPv4 address. One of the goals of this work is to observe the trends regarding the adoption of SPF, DMARC, ADSP and additionally detect the different tags defined in the records.

The first thought was to make use of a database and only save the differences between the distinct measurements. This approach has proven to be inapplicable for a large amount of data due to several reasons. The database became sluggish and sometimes even unresponsive when executing simple SQL queries. It was not possible to filter the data just by using SQL queries what meant that additional processing had to be done. These

[12] https://unbound.net/. Retrieved: 2015-03-12.

[13] http://www.dnssec.net/. Retrieved: 2015-03-12.

were the decisive reasons for leaving this path and continue searching for other solutions which could be able to provide satisfactory results.

All three [1], [3], [4] RFCs describe with the help of ABNF [6] (Augmented Backus Naur Form) their corresponding tags, possible values, separators and distinct syntactic rules. Hence, this creates the opportunity to define regular expression based on the corresponding ABNF. With the help of regular expressions we can break down the TXT records in their individual components, thus determining all the existing tag-value combinations. The outcome would be a CSV file which would contain all possible tags as column names and as cell values: 1 or the number of occurrences if the tag-value combination is available and valid, otherwise 0.

To accomplish this assignment a better understanding of Regular Expressions is necessary. This has been achieved by going through Pythons "re" module documentation[14], a number of books [7], [8] and different web resources[15] which war used to generate[16] and test[17,18,19] the constructed Regular Expressions. Some of the defined Regular Expressions posed no challenge at all while others were rather demanding an obscure looking. Listing 2 shows the difference between the Regular Expression for the version (mprefix) and the IPv4 (mIp4) tag.

```
# version tag
mprefix = re.compile(r"^v=spf1$", re.I)

# IPV4 address mechanism
mIp4 = re.compile(r"^[-+~?]?ip4:(?:(?:25[0-5]|2[0-4][0-9]
|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9]
[0-9]?)(?:\/3[0-2]|\/[1-2]?[0-9])?$", re.I)
```

Listing 2. Regular Expressions Example

After having defined all regular expressions the next step is to define a function which would split the TXT record into different substrings using the "separator" (white space/semicolon) as a splitter; the function would return a python list containing all the substrings. This undertaking was not trivial due to the rules which had to be taken in consideration, like:

- Everything which follows after the "all" mechanism should be dismissed however the "exp" mechanism SHOULD succeed the "all" mechanism but it could also precede it (cf. [1]).
- A TXT record can be composed of multiple strings which should be concatenated before proceeding with its evaluation (cf. [1]).
- Adding a semicolon at the end of a record is optional (cf. [4]).

[14] https://docs.python.org/3.3/library/re.html. Retrieved: 2015-03-12.
[15] http://www.regular-expressions.info/. Retrieved: 2015-03-12.
[16] http://www.msweet.org/abnf.php. Retrieved: 2015-03-12.
[17] https://regex101.com/#python. Retrieved: 2015-03-12.
[18] https://www.debuggex.com/. Retrieved: 2015-03-12.
[19] http://www.regexr.com/. Retrieved: 2015-03-12.

The next step consisted in working out a function which would return the number of tag-value combinations, to assert the existence of SPF, DMARC, ADSP and there validity and also provide the faulty substrings together with an error message. This was achieved by looping over the list which contained the preprocessed TXT record and matching every Regular Expression against every member of the list. For a record to be valid every element of the list had to match a Regular Expression and also convey to a set of rules like: The "v" tag must always be at the beginning of a record and must only be present one time (cf. [1], [4]); the "p" tag must always precede the "v" tag (cf. [4]); the "exp" and "redirect" tags are allowed to appear only one time in the record (cf. [1]); no duplicate tags are allowed to appear in a DMARC record (cf. [4]). This led to three (*get_spf_mechanism*; *get_dmarc_tags*; *get_adsp_tags*) distinct functions which take as parameters the Domain Name and the TXT record.

The work concluded with three Python scripts (*spfTagValue.py*, *dmarcTagValue.py*, *adspTagValue.py*) which are able to provide a separate evaluation for all three (SPF, DMARC, ADSP) record types. A fourth Python script (trinity.py) has been written which is able to aggregate the results of the previous three in one single CSV file. The thus generated CSV files contain 985381 rows and 51 columns. Additional testing was made by using Kitterman's[20] SPF record testing tool and the tools offered by dmarcian[21] for verifying SPF and DMARC.

Further processing was done using the statistics suite GNU R through an installation of RStudio Server[22]. Several R scripts were written which implement the same blueprint, namely: Read all CSV files into R's data frame data structure and apply a function which runs filters on every data frame, returning a list; these lists get merged into a new data frame which gets written into a CSV file. For reading in the CSV files the "data.table"[23] (fread function) package was used due to its superior performance in comparison to the build in function, while for the graphics the "ggplot2"[24], "reshape2"[25], "scales"[26] and "grid"[27] packages were used. Listing 3 shows the loop in which the CSV files are being read and processed thus leading to a data frame which contains the result.

```
for(i in 1:132){
```

[20] `http://www.kitterman.com/spf/validate.html`. Retrieved: 2015-03-12.

[21] `https://dmarcian.com/dmarc-inspector/gabrielkovacs.net`. Retrieved: 2015-03-12.

[22] `http://www.rstudio.com/products/rstudio/`. Retrieved: 2015-03-12.

[23] `http://cran.r-project.org/web/packages/data.table/index.html`. Retrieved: 2015-03-12.

[24] `http://cran.r-project.org/web/packages/ggplot2/index.html`. Retrieved: 2015-03-12.

[25] `http://cran.r-project.org/web/packages/reshape2/index.html`. Retrieved: 2015-03-12.

[26] `http://cran.r-project.org/web/packages/scales/index.html`. Retrieved: 2015-03-12.

[27] `https://stat.ethz.ch/R-manual/R-patched/library/grid/html/grid-package.html`. Retrieved: 2015-03-12.

```

  readRecord = fread(theRecords[i], header=TRUE)
  recordName = theRecords[i]
  processedData = getPercentageData(readRecord, recordName)
  percentageData[i,]= processedData
}
```

Listing 3. R processing loop

The charts and tables in the next section are based on data gathered between April 19th, 2014 and February 19th, 2015 (10 months period). For the progress representation of SPF, ADSP and DMARC a subset of the above data has been generated corresponding to a measurement from every Tuesday, Thursday and Saturday. Starting hour of each measurement being around 8am, CET (Central European Time).

6 Results

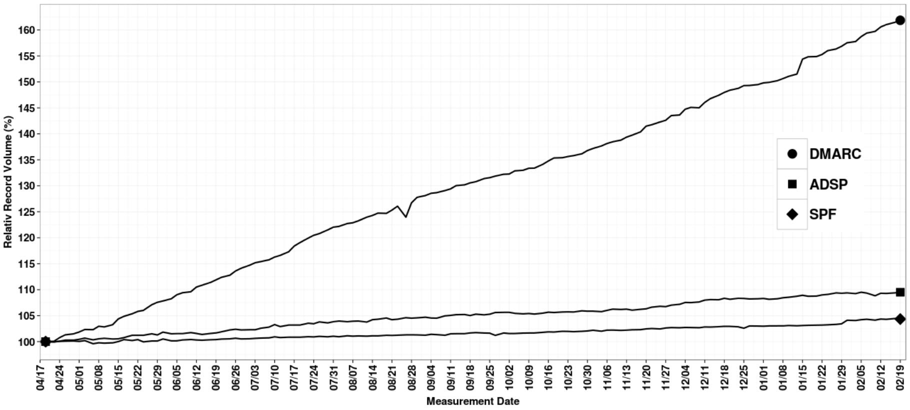

Fig. 2. Development of the number of DMARC, ADSP and SPF entries from measurement M001 to measurement M132.

Figure 2 shows the relative growth of SPF, ADSP and DMARC having as starting point April 19th, 2014 (measurement M001) and end point February 19th, 2015 (measurement M132). The highest growth is registered by DMARC (61.86%) followed by ADSP (9.51%) and lastly SPF (4,35%). It is unclear why ADSP is still being used considering the IETF discouraged[28] its use and that it could easily be replaced by DMARC. Table 1 presents the absolute numbers regarding the deployment of DMARC,

[28] `https://datatracker.ietf.org/doc/status-change-adsp-rfc5617-to-historic/`. Retrieved: 2015-03-12.

Table 1. DMARC, ADSP and SPF entries for the first (M001) and the last (M132) measurement.

Measurement	**DMARC**		**ADSP**		**SPF**	
M001	**Total**		**Total**		**Total**	
	3985 (≈0.40%)		3101 (≈0.31%)		357502 (≈36.26%)	
	Valid	**Invalid**	**Valid**	**Invalid**	**Valid**	**Invalid**
	3757	228	2750	351	349139	8363
M132	**Total**		**Total**		**Total**	
	6450 (≈0.65%)		3396 (≈0.34%)		373071 (≈37.84%)	
	Valid	**Invalid**	**Valid**	**Invalid**	**Valid**	**Invalid**
	6092	358	2996	400	364642	8429

ADSP and SPF for the first and last measurement based on the Alexa list. We can observe that a little bit over a third of the domains defined an SPF record whereas the occurrence of DMARC is very modest. DMARC is not yet a Standard Track within the IETF which could explain why some system administrators could still hesitate to make use of a relative new technology. We can also observe that the vast majority of the published records are valid (cf. Section 5). A record is considered invalid if at least one syntactic error could be identified. Some of the most frequently observed syntactic errors in SPF records are:

- Missing separator (white space) between the tag-value combinations.
- Misspelled tags (e.g., "ipv4" vs "ip4").
- Wrong separators between tags and values; using ":" where a "=" is expected and vice-versa.
- Matching tags with discordant values (e.g., include:91.185.206.154). The only possible value for the "include" tag is a FQDN (Fully Qualified Domain Name).
- The presence of more than one "v" tag.

The most frequent syntactic errors found in DMARC records are:

- The lack of "mailto:" in the "rua" and "ruf" tags (e.g., "rua=dmarc_report@qiye.163.com").
- A white space between "mailto:" and the preceding e-mail address.
- The presence of more than one "rua" and/or "ruf" tag.
- The "p" tag does not immediately precede the "v" tag.
- Matching tags with discordant values (e.g., "sp=s").

Although the ADSP record consists of only one tag-value combination following syntactic errors were identified:

- Instead of "unknown"/"all"/"discardable" the tilde symbol has been used (dkim=∼).

– A semicolon precedes the tag-value combination. Such a symbol is not present in the [3] ABNF.

In the case of SPF and DMARC if a TXT record does not start with the "v" (version) tag then no further inquiries are made by the receiving side thus leading to the conclusion that no SPF and/or DMARC record has been defined. Some domains define multiple SPF and/or DMARC records which start with the corresponding version tag. In this case to eliminate any ambiguity the domain is treated as if it had not published any SPF and/or DMARC records. Table 2 shows the number of multiple records found during measurement M132. Some of the domains which published multiple records have managed to correct this avoidable error by publishing just one SPF/DMARC record. One of the most interesting findings regarding multiple DMARC records is showed in Listing 4. It is unclear how the records which contain one or multiple syntactic errors are interpreted semantically at the receiving side. According to the DMARC draft if a record does not contain a valid "p" (policy) tag but does contain a "rua" (feedback) tag processing should continue with the assertion of a "p=none" policy. This solution is rather counterproductive because it will certainly not be enforced (see: "p" tag MUST follow the "v" tag) by every MTA (Mail Transfer Agent) thus leading to ambiguity.

Table 2. Total DMARC, ADSP and SPF multiple records for the first (M01) and last (M132) measurement.

Measurement	DMARC	ADSP	SPF
M001	6	4	9067
M132	17	4	10457

```
$ dig _dmarc.uzmarketing.com TXT +short
"v=DMARC1\; p=none\; rua=mailto:postmaster@uzmarketing.com"
"v=DMARC1\; p=quarantine\; pct=5\; rua=mailto:postmaster@uzmarketing.com
"v=DMARC1\; p=reject\; rua=mailto:postmaster@your_domain.com, mailto:
    dmarc@uzmarketing.com"
```

Listing 4. DMARC multiple records example

Table 3 shows the absolute and relative figures regarding the distribution of the different SPF policies for M001 and M132 based just on the valid entries. While DMARC and ADSP only provide three different policies SPF offers four. The "+all" mechanism is used by less than 1% (and stagnating) which is still too high considering it authorizes everybody to send e-mails on behalf of the domains who use this policy. The "+all" mechanism can be considered an invitation for spamers to abuse ones domain.

Table 3. Distribution of the SPF policies for the first (M001) and the last (M132) measurement.

Measurement	**"+all"**	**"?all"**	**"∼all"**	**"-all"**	**implicit "?all"**	**redirect**
M001 **% of Total #**	2404 ≈0.69%	78704 ≈22.51%	188239 ≈53.85%	68353 ≈19.55%	3640 ≈1.04%	8262 ≈2.36%
M132 **% of Total #**	2404 ≈0.66%	68262 ≈18.72%	200115 ≈54.88%	81289 ≈22.29%	3911 ≈1.07%	8661 ≈2.38%

According to (cf. [1], 8.2) the "?all" (neutral) mechanism should be treated as if no SPF record was found/defined and that it offers the user the possibility to test SPF. SPF does not offer a standalone feedback mechanism which limits a user's testing possibilities. Only with the help of a DMARC record would it be possible to get a feedback regarding ones SPF configuration. The "?all" policy could be seen as equivalent to DMARCs "p=none" policy. Yahoo is one of the most prominent e-mail providers which uses "?all" as policy for its SPF record. As seen in Table 3 the neutral policy accounts for roughly 20% of all policies and is the only one which registered a deployment decrease. The "∼all" (soft fail) is the predominant (≈55%) deployed policy for SPF which registered an increase or roughly 1% in the last 10 months. One explanation for its popularity could be due to the same policy being used by e-mail providers like Google, Microsoft and Yandex. The SPF documentation suggests not to drop the e-mail immediately but rather also use other technologies (e.g., Grey-listing) to facilitate a final decision. The corresponding DMARC policy would be "p=quarantine". The most restrictive and straightforward SPF policy is "-all". If the sending server is not authorized then the receiving sender should reject the e-mail. Due to the problems SPF has with relaying the use of such a policy could lead to legitimate e-mails being rejected. Surprisingly this policy has the highest growth rate of approximately 2.74% thus making out roughly 22.29% of the total policies.

SPF records which do not explicitly define a policy and also do not contain a "redirect" tag, default (implicit "?all") to a "?all" policy however if the record contains a "redirect" tag then its policy could be found in the SPF record belonging to the referenced domain by the "redirect" tag.

As Table 4 shows the most predominant (≈72.44%) policy for DMARC is "p=none" which is also the only one to have registered a growth of 2.7% to the detriment of "p=reject" and "p=quarantine". This is the policy which should be used when deploying DMARC for the first time because it allows the user to get a better understanding of DMARC without fearing any unforeseen consequences, before moving on to a stricter policy. It is surprising to see that "p=quarantine" only accounts for roughly 8% (and stagnating) of the total policies considering that it would be the next natural step after “p=none” and that in the case of SPF the most common used policy is the middle one. The strictest policy DMARC has to offer, “p=reject” has been adopted by almost 20% of its users registering the sharpest decrease of 2.42%. This policy can be used without any concerns only if a domain never sends e-mails. Yahoo is one of the companies which use “p=reject” while Microsoft and Yandex use “p=none”. Google uses “p=quarantine” for its main (google.com) domain and “p=reject” for its secondary (e.g., google.at) domains.

Table 4. Distribution of the DMARC policies for the first (M001) and the last (M132) measurement.

Measurement	**“p=none”**	**“p=quarantine”**	**“p=reject”**
M001 **% of Total #**	2620 ≈69.74%	314 ≈8.35%	823 ≈21.91%
M132 **% of Total #**	4413 ≈72.44%	492 ≈8.07%	1187 ≈19.49%

Table 5. DMARC domain and subdomain policy distribution for M132.

Domain policy	**Subdomain policy**			
“p=none"	**“sp=none”**	**“sp=quarantine”**	**“sp=reject”**	**“no sp”**
4413	16.36%	0.23%	3.81%	79.60%
“p=quarantine"	**“sp=none”**	**“sp=quarantine”**	**“sp=reject”**	**“no sp”**
492	9.15%	9.35%	2.64%	78.86%
“p=reject"	**“sp=none”**	**“sp=quarantine”**	**“sp=reject”**	**“no sp”**
1187	3.88%	0.17%	8.68%	87.27%

Table 5 shows the distribution of the explicit subdomain policy (*sp* tag) for every distinct organizational (main) domain policy (*p* tag) contained in the DMARC records for the last measurement (M132). If no “sp” tag is defined the subdomain policy will correspond to the value present in the “p” tag thus defining a “sp” tag with the same value as the “p” tag is redundant. However a subdomain can have its own DMARC record thus overriding the previous policy defined in the upper (organizational domain) level. When a subdomain has its own DMARC record it does not pass its policy down to the subsequent subdomains due to the way how the DMARC check is done. For example: If the subdomain a.b.example.com is checked for a DMARC record and it does not define one then the next step is to check the example.com domain and not b.example.com. This measure has been taken so that the DNS traffic can be kept to a minimum. The majority of the published DMARC records do not make use of the “sp” tag however we can also observe records containing identical “p” and “sp” tags. It seems that the users of the “p=reject” policy have a better understanding of the DMARC policy mechanics. A “p=none” and “sp=reject” combination could suggest that e-mails are being sent only from the main domain while a “p=reject” and “sp=none” combination could lead us to believe that e-mails are being sent only from the subdomains.

ADSPs policy distribution and trend presented in Table 6 behave as expected. The share of “dkim=unknown” makes out most (≈64.99%, M132) of the ADSP records while “dkim=discardable” only represents about 5% of the total. Both policies registered a decrease in the favor of the “dkim=all” (≈30.17%, M132) policy which allows the user to make a statement without taking a big risk and the same time aid the receiver in its decision making. Although DMARC maps ADSPs policies one to one at the time being the development of DMARC does not seem to follow ADSPs one.

Table 6. Distribution of the ADSP policies for the first (M001) and the last (M132) measurement.

Measurement	“dkim=unknown”	“p=all”	“p=discardable”
M001 **% of Total #**	1838 ≈66.83%	772 ≈28.08%	140 ≈5.09%
M132 **% of Total #**	1947 ≈64.99%	904 ≈30.17%	145 ≈4.84%

The data presented in Table 7 shows the absolute numbers as well as the relative distribution for the minimalist SPF records. It is unclear why someone would publish an “+all” record considering it can do more harm than good. The “+all” record usually indicates that its user might be a spamer and it is also an invitation for spamers to abuse a domain name. This record should never be published instead the use of the “?all” policy is encouraged. This record is beneficial for those who do not wish to have their outgoing e-mails filtered on the receiving side based on SPF. For big companies who run their own name servers and with a high volume of outgoing e-mails such a record with a high TTL can reduce the number of queries their name servers have to process. Defining a “~all” record is detrimental for the sender as well as for the receiver. Due to the absence of an IP address which would authorize the sending server all incoming e-mail would be treated as suspicious e-mail which would increase on the receiving side the need of additional processing. This could lead to legitimate e-mails being rejected (worst case) or to be delivered to the

Spam folder which is surely not in the best interest of the sender. If a domain never sends e-mail the best way to protect its reputation is to use a “v=spf1 -all” record. This sends a clear message to the receiving server which can reject the e-mail during the SMTP connection, not having to waste further valuable resources with the e-mail processing. It is encouraging to see that the absolute number of “-all” records has almost doubled while the other ones are rather stagnating. Google publishes such a SPF record for all of its secondary domains (e.g., google.at).

Table 7. Number of minimalist SPF records of the type “v=spf1 (+/?/~/-)all”.

Measurement	“+all”	“?all”	“~all”	“-all”
M001 **% of Total #**	457 ≈19.20%	244 ≈10.25%	94 ≈3,95%	1585 ≈66.60%
M132 **% of Total #**	457 ≈11.64%	228 ≈5.81%	103 ≈2.62%	3137 ≈79.93%

Table 8 shows the occurrence of the minimalist DMARC records for M001 and M132. While the SPF trends regarding the minimalist records was clearly towards the most restrictive policy in the case of DMARC however a shifting away from “p=reject” can be noticed favoring the “p=none” (neutral) policy. A SPF record of type “v=spf1 -all” combined with a “v=DMARC1; p=reject” record can additionally enforce a domain owners statement that it does not send e-mails and reassures every receiving servers to drop e-mails claiming to be from the domain which implements such strict policies. However using such a strict policy in case of a domain which dose send e-mail is discouraged because this could lead to legitimate e-mail not being delivered. This could explain the shifting from the strictest policy towards a safer one like “none” or “quarantine”. Unfortunately the lack of a “rua” tag (used for reporting) makes debugging the possible problems rather difficult if not impossible. The “p=none” policy also offers the same benefits as the definition of a “v=spf1 ?all” record. Using the “v=DMARC1; p=quarantine” record can protect a domain against spoofers by raising the attention on the receiving side in case SPF and/or DKIM would fail to confirm the sending server.

Rather than looking at each technology separately Tables 9 and 10 provide a view which is meant to determine the correlation between the policies of SPF vs. DAMRC and ADSP vs. DMARC. Table 9 presents the distribution of the SPF records based on their policy (“?/~/-all”) and their associated DMARC records with their respective policies. The domains which use “?all” as a SPF policy also align their DMARCs records policy by using “p=none”. This behavior can also be observed between

Table 8. Number of minimalist DMARC records of the type "v=DMARC1; p=(none/quarantine/reject)".

Measurement	**"p=none"**	**"p=quarantine"**	**"p=reject"**
M001 **% of Total #**	42 ≈27.45%	33 ≈21.56%	78 ≈50.99%
M132 **% of Total #**	127 ≈43.20%	64 ≈21.77%	103 ≈35.03%

Table 9. SPF records and corresponding DMARC policies.

Measurement	DMARC/SPF	DMARC-Policies		
	"?all"	"p=none"	"p=quarantine"	"p=reject"
M01	202 out of 78704	77.22%	14.36%	8.42%
M132	315 out of 68262	88.25%	4.13%	7.62%
	"∼all"	"p=none"	"p=quarantine"	"p=reject"
M01	1561 out of 188239	80.14%	7.30%	12.56%
M132	2685 out of 200115	81.19%	8.23%	10.58%
	"-all"	"p=none"	"p=quarantine"	"p=reject"
M01	1618 out of 68353	57.66%	10.88%	31.46%
M132	2496 out of 81289	59.66%	10.13%	30.21%

Table 10. ADSP records and corresponding DMARC policies.

Measurement	DMARC/ADSP	DMARC-Policies		
	"dkim=unknown"	"p=none"	"p=quarantine"	"p=reject"
M01	182 out of 1838	74.17%	23.08%	2.75%
M132	248 out of 1947	76.61%	19.35%	4.04%
	"dkim=all"	"p=none"	"p=quarantine"	"p=reject"
M01	137 out of 772	50.36%	21.90%	27.74%
M132	215 out of 904	48.84%	21.40%	29.78%
	"dkim=discardable"	"p=none"	"p=quarantine"	"p=reject"
M01	80 out of 140	3.75%	13.75%	82.5%
M132	91 out of 145	13.19%	13.19%	73.62%

"-all" and DMARCs stricter policies like "p=quarantine" and "p=reject" although not being as categorical as in the case of "?all" and "p=none". The correlation between "∼all" and "p=quarantine" is rather weak. Table 10 shows all ADSP records and their corresponding DMARC records. The alignment between the ADSP "dkim=unknown" and "dkim=discardable" policies and their DMARC equivalent "p=none" and "p=reject" is as expected relative strict whereas in the case of "dkim=all" the alignment to “p=quarantine” is rather loose and a tendency towards the “p=reject” policy can be observed.

Unfortunately it was not possible to measure the deployment of DKIM which could have explained why in the case of DMARC the “p=quarantine” policy is omitted in favor of a much stricter policy like “p=reject”. The majority of domains which define a SPF and DMARC record tend to coordinate their policies although even if seldom there are domains which publish nonsense combinations like a SPF “+all” policy together with a DMARC “p=reject”. In the absence of DKIM even a “?all” and “p=reject” combination could be questionable.

7 Conclusion

7.1 Summary

This work presented the fundamentals of some of the technologies (SPF, DKIM, ADSP, DMARC) which have been developed to make e-mail authentication possible thus providing means to mitigate the threats of spam and phishing. It also investigated the adoption rates and configuration of the above mentioned technologies. For this purpose a large scale measurement (based on the Alexa Top 1 million list) has been conducted between April, 2014 and February, 2015.

As the observations of our large-scale measurement illustrated, more than a third of the domains contained in the Alexa Top 1 million list have made use of an SPF record. There are still new domains deploying SPF, while the development has been rather flat during our measurement period. DMARC has registered a very high adoption rate in relative numbers, but in absolute terms is far off the high deployment standards of SPF. Considering SPF has been present since more than nine years (cf. [9]) while DMARC is new and vitally developing, it appears without a doubt that DMARC will establish itself. Companies like Agari and ReturnPath have successfully started since at least 2013 to offer paid services based around DMARC for large e-mail providers, banks and other financial institutions.

DMARC was able to avoid some of SPFs shortcomings by defining a cleaner syntax, not having a “+all” equivalent policy and also a better naming choice for their policies (“soft fail” vs “quarantine”). It is unclear why the DMARC record should be published under a subdomain and not the main domain considering that the SPF record is defined under the main domain and thus a single query would suffice to acquire both records. At the time being there is no free solution for the interpretation of the DMARC feedback. While for a single user or small company this may not be a problem, as they may not require any DMARC analysis results themselves, it does pose a financial burden for big companies. Microsoft (outlook.com) relies on commercial services such as from Agari and ReturnPath for the feedback evaluation. DMARC is still a relative new technology that will have time to mature and to prove its worth. It is encouraging to see that so much effort is put in increasing e-mail security and at the same time making it less resource dependent.

7.2 Future Work

Using the static Alexa Top 1 million list provided an insight regarding the deployment of SPF, ADSP and DMARC. However the *www* is a very dynamic place where every day new domains are being created, change owners or are dying. This is why we wish to expand the measurements based on the Zone Files obtainable from ICANN and Verisign (.com, .net) thus enabling us to make a better estimation regarding the adoption of DMARC and SPF. At the time being we are not looking up the SPF records which can be found under the domains defined by the "include" and "redirect" tag. We would like to expand the script to allow us to completely evaluate SPF in the future. At the moment we are able to detect the errors in the records but do not provide a solution to fixing configuration problem. This is also a path which we would like to explore in order to improve the quality of DMARC and SPF records.

By checking the TXT RR it is only possible to determine the adoption rate and the policies defined by DMARC and SPF. It would be interesting to conduct a study to see how DMARC and SPF impact e-mail traffic when in use and if these technologies have the potential to be a replacement for the previous existing technologies.

As previously mentioned at this time there is no freely available software for the DMARC feedback processing. Building such a software could aid those who cannot afford or are not yet willing to pay for such a service. There would also be no need the share sensible e-mail information with others.

References

[1] S. Kitterman, "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1," RFC 7208 (Proposed Standard), Internet Engineering Task Force, Apr. 2014, updated by RFC 7372. [Online]. Available: http://www.ietf.org/rfc/rfc7208.txt

[2] D. Crocker, T. Hansen, and M. Kucherawy, "DomainKeys Identified Mail (DKIM) Signatures," RFC 6376 (Internet Standard), Internet Engineering Task Force, Sep. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6376.txt

[3] E. Allman, J. Fenton, M. Delany, and J. Levine, "DomainKeys Identified Mail (DKIM) Author Domain Signing Practices (ADSP)," RFC 5617 (Historic), Internet Engineering Task Force, Aug. 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5617.txt

[4] M. Kucherawy and E. Zwicky, "Domain-based Message Authentication, Reporting and Conformance (DMARC), Ver. 04," Internet Draft, Internet Engineering Task Force, Apr. 2014. [Online]. Available: http://tools.ietf.org/html/draft-kucherawy-dmarc-base-04

[5] M. Andrews, "Negative Caching of DNS Queries (DNS NCACHE)," RFC 2308 (Proposed Standard), Internet Engineering Task Force, Mar. 1998, updated by RFCs 4035, 4033, 4034, 6604. [Online]. Available: http://www.ietf.org/rfc/rfc2308.txt

[6] D. Crocker and P. Overell, "Augmented BNF for Syntax Specifications: ABNF," RFC 5234 (Internet Standard), Internet Engineering Task Force, Jan. 2008, updated by RFC 7405. [Online]. Available: http://www.ietf.org/rfc/rfc5234.txt

[7] S. Levithan and J. Goyvaerts, *Regular expressions cookbook*, 2nd ed. O'Reilly Media, Incorporated, 2012.

[8] J. Friedl, *Mastering regular expressions*, 3rd ed. O'Reilly Media, Incorporated, 2006.

[9] M. Wong and W. Schlitt, "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1," RFC 4408 (Experimental), Internet Engineering Task Force, Apr. 2006, obsoleted by RFC 7208, updated by RFC 6652. [Online]. Available: http://www.ietf.org/rfc/rfc4408.txt