University of Amsterdam

Research project 2

Designing an open source DMARC aggregation tool

Yadvir Singh

supervised by
Michiel Leenaars

August 17, 2016

Abstract

DMARC provides a standard for interaction between a domain owner and the recipient of an email. In this interaction, the domain owner is eligible to receive DMARC reports that contain various information about validation results of emails. This research investigates the various information elements that can be extracted from DMARC reports and from sources surrounding DMARC . Several additions are proposed to already available commercial tooling that can parse and processes DMARC data. On top of the proposed additions, an open source implementation is built that aims to aid in the initial deployment and monitoring of DMARC.

Acknowledgements

I would like to thank Michiel Leenaars for his support during the research. His contributions provided really helpful insight into various elements of DMARC and with development of additional tools.

Contents

1 Introduction

A long lasting problem with email are the unauthorized messages sent on behalf of a domain. Technologies like SPF (Sender Policy Framework) [1] and DKIM (DomainKeys Identified Mail) [2] try to combat such problems with the usage of email validation. Although both technologies provide protection against unauthorized email, there is no interaction between the recipient and the legitimate owner of a domain apart from the retrieval of the DNS TXT records of DKIM and SPF. If either the SPF or DKIM check fail, its up to the receiver to decide which action must follow and the sender has no automated way to find out whether the email is kept or thrown away. DMARC (Domain Message Authentication Reporting & Conformance) [3] provides the missing interaction between the receiver and the domain owner. The domain owner can publish a policy that guides the recipient on what actions should be taken. Besides the exchange of policy, DMARC also provides a mechanism to receive feedback on authentication results generated at foreign domains. Feedback is sent to the domain owner in the form of aggregate or forensics reports that among other information shows the results of the DKIM and SPF checks as well as the IP address of the sender of an email. The authentication results present in these reports can be used to evaluate the security setup of a domain. Also, reports can be used for monitoring a domain to see if its currently being spoofed by some entity.

This research will focus on the feedback reports that are sent to the domain together with other information elements that surround DMARC . These reports are analyzed, parsed and aggregated to give a complete overview of the current (DMARC) domain status. The conversion from DMARC reports to a user friendly overview is not a new concept. Together with the increasing number of domains that deploy DMARC, several commercial parties have developed tooling that parse and visualize DMARC data. This paper analyses a number of commercial tools, as well as producing a gap analysis of missing features in these tools based on the information sources available to the domain owner. Part of the proposed features are implemented in a newly developed open source tool which is made available freely.
The main research question is:

> *How can aggregated DMARC reports provide domain administrators insight into their email domain?*

Subquestions include:

1. How can DMARC feedback be used to aid in the deployment of DMARC?
2. How can DMARC feedback be used to monitor spoofing?
3. What information can outgoing DMARC reports provide to the domain owner?

Section 2 will give a brief overview of DMARC and its workings. Section 3 will review possible information sources that can be used for analysis and monitoring purposes. These include sources already used by available tools, as well as unused sources. Next, section 4 will discuss some DMARC tools developed by commercial parties. Section 5 continues with the proposal of additional tools and instruments that are not yet implemented in the reviewed tools. In section 6 an open source implementation is discussed which incorporates some of the proposed tooling. Section 7 will end with the conclusion followed by proposed future work.

2 DMARC

This section will briefly discuss the workings of DMARC followed by a dissection of the feedback reports.

2.1 General workings

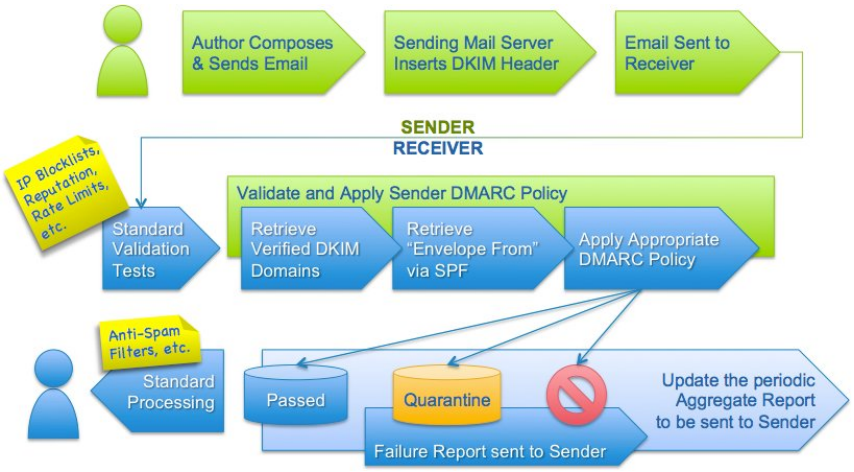

Figure 1: DMARC processing and authentication.[4]

To combat unauthorized email, one can use SPF and DKIM. Both provide ways to validate received email. In SPF, a domain owner can publish a record that contains IP address(es) that are authorized to send mail on behalf of the domain. Whilst SPF only checks the sender, DKIM can also validate the authenticity of the message itself. Using the public key of the legitimate domain, a receiver can check if the message is indeed signed by the legitimate key holder. If either SPF or DKIM fails, its up to the receiver what action must follow. This is where DMARC steps in. Rather than making a local decision, DMARC provides a mechanism for domain administrators to publish a policy that can be adhered to by the recipient. Such a policy indicates how the receiver should handle email (that appears to be) coming from the owner's domain. The domain owner should publish the policy in a DMARC DNS TXT record that. Bellow an example of of DMARC record is given:

```
v=DMARC1; p=none; sp=none; rua=mailto:rua@dmarc-research.nl;
ruf=mailto:ruf@dmarc-research.nl; rf=afrf; pct=100; ri=86400
```

The first (`v=DMARC1`) and mandatory parameter specifies the version of DMARC. The second parameter defines the policy that is to be adhered by foreign domains. This parameter specifies what policy should be applied when the message fails the DMARC check. This value can be set to three options: none, quarantine or reject. If set to none, no actions should be taken. When the policy is set to quarantine, the message should not be discarded but put aside for later evaluation. Lastly, when the policy is set to reject, the message should be discarded. The `sp` parameter fulfills the same functionality for sub domains. Both when the DMARC check fails or succeeds, the domain owner can ask to receive reports. The address to which these reports are to be sent is specified by the `rua` and `ruf` parameters. The `rua` specifies the email address to which aggregate reports must be sent. These reports contain aggregate authentication results that are collected over a certain period of time. The `ruf` parameter functions similarly for forensics reports. Forensic reports contain more specific details about emails such as the subject line and email header information of the evaluated email. The `rf` parameter specifies the format of forensic reports that the owner wishes to receive. The default value of this parameter is set to `afrf` (Authentication Failure Reporting Format).[5] Earlier drafts on DMARC also included the IODEF (Incident Object Description Exchange Format) format as an option.[6, 7] The `ri` parameter advertises the interval (in seconds) in which the domain owner wishes to receive reports. For the full specification and parameter list, please refer to RFC 7489.[3] For a visual representation of the workings of DMARC see figure 1.

```
<report_metadata>
<org_name>acme.com</org_name>
<email>noreply-dmarc-support@acme.com</email>
<extra_contact_info>http://acme.com/dmarc/support</extra_contact_info>
<report_id>9391651994964116463</report_id>
<date_range>
        <begin>1335571200</begin>
        <end>1335657599</end>
</date_range>
</report_metadata>
```

Listing 1: DMARC report meta data extract.

```
<row>
<source_ip>192.0.0.1</source_ip>
<count>2</count>
<policy_evaluated>
        <disposition>none</disposition>
        <dkim>fail</dkim>
        <spf>pass</spf>
</policy_evaluated>
</row>
```

Listing 2: DMARC report authentication results extract.

2.2 Aggregate reports

Aggregate reports are XML formatted documents that contain various elements that contain authentication results and information about the domain from which the report originates. These reports are contained within a ZIP archive that is usually send every 24 hours.
Listing 1 shows part of a DMARC XML report containing meta data about the origin of the report. This section contains the name and email of the organization that sent this report. To uniquely identify reports, each report is marked with a unique ID. The `data_range` tag specifies the time over which the information in this report is collected. This time span is usually a 24 hour period.
Listing 2 shows part of the report which contains a summary of the authentication results. This includes an IP address found in the received email. Following the `source_ip` is the `count` tag, which represents the number of aggregate results to which these results apply. The `disposition` tag shows which policy was applied. Finally, `dkim` and `spf` tags show the result of the corresponding check. For a detailed evaluation of these fields, see section 3.

3 Data sources

Various information can be extracted from incoming DMARC data and other related sources. This section will discuss four sources from which data can be extracted. These include: DMARC reports, Mail envelope and DNS information as well as external sources.

DMARC reports provide the primary source of information. The first section of a report contains information about the sender of the report and some additional meta data. The first field in this section contains the organizational name of the source. Although this can be set to the domain from which the reports is send, its not mandatory. It can for example, also hold the company name.

The second element that can be used in analysis is the report ID. This value is meant to uniquely identify each report. The RFC on DMARC [3] contains no recommendation on how this ID should be constructed. Therefore, different sources can apply different methods for generating the ID. Some domains can use increments while others correlate a time stamp to the report ID. Some providers will use the arrival time of the first email that is checked of a particular domain as the report ID. When only one email is collected during a time span, the exact time of the message can in some cases be extracted from the report ID.

The last field from the meta data section that can be used for analysis is the `data_range` field. This field contains the date range over which the results in the report are collected. This time span is not a fixed value, however, the RFC on DMARC [3] strongly suggests a common time period or correlation of multiple sources becomes difficult or impossible. This times span is often set to a period of 24 hours. In cases (when a value other than 24 hours is chosen) this might lead to information on specific aspects of a DMARC setup such as the use of a certain vendor or service.

The authentication summary of DMARC reports contains various information elements that can be used in analysis. The first element of interest is the source IP address. This address denotes the source of the message(s) that have been checked by the report generator. Third party sources like the Spamhaus Block List [8] can be used to determine if the IP address is known to send illegitimate email.

The authentication section also provides the result of SPF and DKIM checks performed on messages. Coupled to the authentication results is a counter that denotes the number of messages to which this authentication result applies. Together, these information elements can give information about the number of emails that passed or failed SPF and/or DKIM and the total number of emails that have been processed.

Besides aggregate reports, DMARC can also provide a domain owner with forensic reports. These reports contain more detailed information about individual emails that have been evaluated by the report generator. By default the reports are composed according to the AFRF specification.[5] Like aggregate reports, forensics reports contain the IP address of the original sender of the evaluated message and the DKIM and SPF authentication results. Forensics reports may also contain the email envelope From address, Mail From address, original envelope-ID and the time at which the message arrived at the report generator. Additionally, the subject line of the original message can be included in these report. Listing 3 shows an example of a forensics report.

Aside from information obtained by from DMARC reports, the email in which these reports are sent can also provide supplementary information. An email header is composed of several parts which include information about the sender. By reviewing these fields, information about the sender of the DMARC reports can be extracted. One of the elements found in the email header is the source IP address of the sender which can be extracted from the `Received` header field. Another element is the mail From address that holds the senders mail address. These elements can be useful for tracking incoming reports and their sources. The same header field can also reveal the path the email has taken whilst in transit from sender to the recipient.

Additionally, email headers can also contain SPAM scores from various tools (like SpamAssassin). Besides SPAM checks, the email can also undergo checking by virus scanners that will report the result in the email header as well. These scores can help with the determination of the legitimacy of an email that contains the report(s).

Besides information extraction from incoming sources, one can also evaluate the domain status by reviewing the DNS records of the domain. This includes the DKIM , SPF and DMARC records. Such

information can be used to analyze the current state of a domain and by recording these records, changes made to the record(s) can be reviewed and correlated to changes in the mail flows - or more specifically, to the reports describing these mail flows.

Next to receiving reports, a DMARC enabled domain typically (although not mandatory) sends reports. These reports contain authentication results of incoming emails at the monitored domain. This information can be of interest as it shows the volumes of email that pass or fail DMARC checks, their sources and the domain which they might try to spoof.

Table 1 summarizes the information that can be extracted from the mentioned sources.

DMARC Aggregate reports	DMARC Forensics reports	Email Headers	Miscellaneous
• Reporting Org. Name	• Source IP of message	• Source IP of reporter	• Current DMARC , SPF and DKIM configuration (from DNS data)
• Report ID	• SPF , DKIM Result	• Path of message (hops)	• Previous configurations of SPF , DKIM and DMARC
• Date Range	• From address	• SPAM scores	• Third party sources (SPAM blacklists)
• Source IP of message	• Mail from address	• Various test results (anit-spam, antivirus)	• Incoming reports
• SPF , DKIM Result	• Original envelope ID	• Mail From address	
• Message volume	• Time of received message at reporter		
	• Subject line of message		

Table 1: Table showing the various information elements that can be extracted from DMARC reports and surrounding sources.

4 Comparative analysis of DMARC services

Several commercial parties offer tools that can process DMARC reports. These services target companies that do not have the capability or expertise to process reports in house. When a company decides to use such services, it is often provided with a special email address that must be published in the DMARC record (`rua` and `ruf` parameters) of the domain. This email address will forward all reports to the commercial party for processing. This section reviews the current implemented features in three commercial tools. The evaluation is based on public sales information and documentation from each vendor. Table 2 summarizes the findings.

	Dmarcian [9]	DMARC analyzer [10]	Postmark [11]
DNS record creator	✓	✓	✗
Categorization	✓	✓	✓
Current configuration status	✓	✓	✗
Time graphs	✓	✓	✗
Forensic report analysis	✓	✓	✗
Whois IP lookup	✗	✗	✓
Detailed Filters	✗	✓	✗
Map based Visualization	✓	✗	✗

Table 2: Comparison of implemented features in commercial tooling.

Most of the commercial parties provide the user with a DNS record creation tool. These tools serve the purpose of helping domain owners creating an appropriate TXT DMARC record. A DMARC record can consist of multiple parameters that must be configured correctly. A domain owner might not have the required level of DMARC knowledge to achieve a correct setup of these parameters. Therefore, a common set of parameters is either pre-configured or additional information is provided so that a domain owner can setup a custom record. Besides explanation of the parameters, the tools also give recommendations on how certain parameters should be setup.

Categorization of the aggregate results is implemented in all three evaluated products. In general, message authentication results are categorized into Trusted and Unknown/Threats messages. Results that fall in the trusted category are messages that were sent from authorized domains (for example specified by SPF). These sources are explicitly allowed to send email on behalf of a domain. Ideally, the domain owner should see all messages that are sent from authorized sources fall in this category. The second category holds messages that did not pass the DMARC check and are flagged as potentially unwanted. This category can consist of emails that are indeed spoofing the monitored domain. However, legitimate email can also end up in this category due to configuration errors of authorized senders. Additional categorization (such a forwarders) is handled differently by different parties.

Additionally commercial tools provide the user with an overview of the current status of their domain. This status information covers DMARC as well as SPF and DKIM . The status is based on the contents of the respective DNS records and incoming reports which can be used to check if SPF and DKIM checks pass or fail from authorized sources. If any configuration or attention is needed, the user will be notified.

Graphs form an important aspect of the reviewed tools. These visualizations enable the user to review authentication results over a certain time span. The analyzed tools allow one to select a data range over which the aggregate authentication results are displayed. This includes individual DMARC results as well as separate SPF and DKIM results. Graphs can be generated for both trusted and untrusted sources.

Another feature that is implemented by Dmarcian and DMARC analyzer is the ability to processes forensic reports. These tools allow the user to inspect forensic reports which can be used to analyze illegitimate mail as well as aid in the correct deployment of DMARC when mail from legitimate sources fails DMARC checks.

One of the features that is found only in Postmarks's DMARC tool is the integration with an IP lookup functionality. In all tools, the user can review the IP address that is extracted from reports. However, the tool from Postmark offers an automatic whois lookup to acquire more information about an IP address. This supplies additional information like the owner of the address, a location approximation and possible contact addresses.

Another feature that is found in both Dmarcian and DMARC analyzer are the implemented filtering options. The degree of filtering differs from vendor to vendor. The tooling from DMARC analyzer provides the user with more extensive options which include filtering on individual DKIM and SPF results as well as filtering on IP address found in reports. This functionality allows one to review aggregate results based on a specific interest.

An alternative visualization is found in the tooling provided by Dmarcian. This visualization maps the IP addresses found in reports to a global world map by retrieving location information from IP address registers. The accuracy of such location information can differ per IP address. Therefore, the visualization only assigns IP address to countries. Each country is color coded to depict the amount of email that falls in the Unknown/Threat category.

All of the features mentioned above rely (mostly) on obtaining information from incoming DMARC reports. Additional sources like the email envelope and statistical measurements (of DNS records) seem to (mostly) be unused in the evaluated tools.

5 Proposed additions

Although tools discussed in section 4 provide decent functionalities to process DMARC data, some additional features can be implemented. This section will discuss some features that are missing or can be improved upon.

5.1 Outgoing reports

When a domain opts for one of the commercial tools, in most cases an email address of the commercial party is published in the domain's DMARC record which will redirect reports to the commercial party directly. Although this enables the domain owner to review incoming reports, outgoing reports are not being processed. Like incoming reports, outgoing reports can provide the domain owner with information about his domain. Similar to incoming reports, these reports contain information about DKIM , SPF alignment and IP source information. By monitoring these reports, the domain owner can get insights into incoming emails and their authenticity. This can be of interest if for example the monitored domain is subjected to a SPAM campaign. If the legitimate domain of the incoming SPAM emails has implemented DMARC correctly, the spoofed email will fail the DMARC check. By monitoring outgoing reports, the domain owner can review if he is under attack by spoofed emails and record the source(s). Also, one gets a grip on the volume of reports that are sent to each domain and also an insight into the diversity of domains to which these reports are sent.

5.2 Automated DMARC tester

Because the reliance of DMARC on SPF and DKIM , the domain owner should not enforce strict DMARC policy before both SPF and DMARC are setup correctly. Therefore a verification or test functionality is proposed that will help determine if DKIM and SPF are setup correctly. The proposed verification tool will use a separate mailbox that will only be used for testing purposes. The automated tester can automatically generate an email that will be send to the reserved mailbox. After receiving the email, the tool will parse and check if the email is SPF and DKIM aligned. This check will be performed independently of the SPF and DKIM software used on the domain owners infrastructure. The result of the checks are automatically presented in a graphical user interface. Such a service is provided by one of the reviewed products. However, this service requires the domain owner to manually send an email to the commercial party after which an email is sent back with the authentication results. The proposed automated solution would not require the domain owner to manually send a test mail and also eliminates the need to manually check the authentication result in the response mail. Moreover, the inability to review the raw contents of the sent test email (to review the DKIM header for example) forms a drawback when using the commercial tool.

5.3 DNS record history

During the deployment or operation of DMARC, the domain owner can make changes to his DMARC, DKIM or SPF record(s). Most of the commercial parties mentioned in section 4 offer graphs that visualize the volume of mails that pass SPF, DKIM , or both over a certain time span. Building on top of these visualizations, a time line with changes to the DNS records is proposed. This time line shows points at which either the SPF, DKIM or DMARC DNS record have been altered. Using this information, a domain owner can track the relative effect of changing DNS records to the number of messages that passed or failed the DMARC check. Besides usage in the initial deployment, this tool can also be used while DMARC is in regular operation.

5.4 AS visualization

Email sent to a particular domain can come from various sources spread across the internet. This diversity translates to different IP address that are found in the DMARC reports. These IP addresses are again linked to different ASs (autonomous systems). This information might be of interest to a domain owner as it shows the relative diversity of sources that send email.

Coupled with this diversity, one can also visualize the authentication results and email volume of autonomous systems. Color markings can represent the ratio of emails that themselves pass SPF and DKIM or fail both. This gives the domain owner a quick overview of the sources that send email on behalf of the domain. This can both be used for monitoring legitimate sources and also sources that possibly try to spoof the monitored domain. If an autonomous system retains a bad rank (that is repetitive failing the SPF and DKIM check) over a certain time period, the domain owner might decide to put IP address mapped to this AS on some form of a blacklist. Additionally, the domain owner can contact the AS administrator for clarification.

5.5 Full text search on reports

Many of the commercial tools convert the incoming XML formatted reports to graphical visualizations with the intention to hide the XML reports. However, one might want to review and extract information from these reports manually. Therefore, a full text search functionality is proposed. This functionality should allow users to work with a generic query language (such as XQuery) to retrieve the desired information from collected reports.

5.6 SPAM check

The failure of SPF and DKIM can have different causes. One of the sources can be a misconfigured mail server that is authorized to send email on behalf of a domain, for which SPF and DKIM are not setup correctly. In this case, a domain owner would be interested in the DMARC reports as they contain the IP address of a potentially misconfigured email server. Another source can indeed be email with the intention to spoof a domain. To check if the monitored domain is being used in a SPAM campaign, third party tools can be used. An example would be the SBL database of Spamhaus [8] which contains IP addresses that are reported to be sending illegitimate email. Besides known addresses from public sources, one can mark addresses that are not in the public databases that might, with some degree of uncertainty, be sending SPAM. These addresses are of interest as they can possibly reveal a specific targeted SPAM attack on the monitored domain.

5.7 DMARC endpoint check

Correctly configured DMARC domains can send reports on a regular basis. However, domains can be setup (unconsciously) incorrect resulting in no feedback reports from this domain. To get an overview of such domains, a DMARC endpoint checker is proposed. This functionality records email exchanged with domains that have published a DMARC record. With the assumption that if a domain has published a DMARC record, it will also generate feedback, the incoming stream of reports is monitored to see if any feedback is received. If no feedback is received over a long time span (for example: at least 2 times the requested interval in the domain owners published DMARC record, with a minimum of 48 hours) the domain owner should be notified. Besides email exchanged with domains that have published a DMARC record, there can also be domains that do not have a DMARC record published. Such domain most likely do not have deployed DMARC. The owner can use this list to scan for important email contacts that should be applying the DMARC policies, and contact their respective administrator with the motivation to deploy DMARC.

6 Implementation

This section will discuss an open source implementation of a DMARC deployment and monitoring tool. Section 6.1 will discuss the motivation behind the need for an open source alternative. Section 6.2 describes the design of the developed tool as well as the setup of the test domain. Sections 6.3 and 6.4 discusses the two tools that have been developed. Section 6.5 continues with an overview of two visualizations that have been implemented. The complete source code can be found at: `https://github.com/cheatas/DMARCate`.

6.1 Motivation

Domain owners that make use of commercial tools like the described in section 4, are bound to redirect reports to these commercial parties which potentially operate under another legal jurisdiction. An alternative that would let domain name owners process reports within their own organization could therefore be preferred. Additionally, the use of open source would allow inspection and customization of the tools for specific purposes unique to the users own needs. This includes integration into existing operational software. Therefore, an open source implementation is build that aims to help in the deployment of DMARC as well as monitoring the domain after initial deployment. Part of the proposed additions are implemented in the developed tooling. Unlike commercial tools, ease of use is not one of the primary goals of the developed open source tooling. The target is to allow domain owners to customize the tooling according their own needs without the need using the entire software suite. This is achieved by a modular design in which tools can be used independent from each other. The cost of this flexible design results however in more manual configuration compared to the commercial tooling.

6.2 Setup

6.2.1 Front end

To offer platform independence, the graphical interface is based on a web application. This allows users to use the tool in a browser that is operational on various devices. The web interface consists of several widgets that operate individually from each other. The widgets are also independently updated. This allows a variation in the update frequency depending on the needs of the user. The independence of the widgets also allows developers to integrate individual widgets within other applications. The developed tool makes use of two open source frameworks. Firstly, the Bootstrap framework [12] is used as a base for the web interface. This framework allows one divide a web page into several blocks in which web content can dynamically scale for mobile usage. Secondly, the Rickshaw library is used for creating graphs.[13] One of the additional benefits of this library is that it features a time line that can show additional information below the graph. The usage of this feature is discussed in section.6.4.4

6.2.2 Back end

The two main parts of the back end are formed by the logic part which renders/updates widgets and the database part which stores incoming reports. The logic part is written in Python and is responsible for retrieving data from the database through SQL queries and is also responsible for generating the HTML files. The individual HTML files are united into one web page that represents the complete graphical interface. To store and search reports, a MySQL (or compatible) database is used. Incoming reports are parsed by Techsneeze's dmarcts-report-parser.[14] Each field of the report is mapped to a column in the database table. Besides storing all values of the report individually, the raw report is also stored.

6.2.3 Domain & software

To test the the various tools developed, a real domain was registered. This allowed to test the functionality against an operational domain and receive real world reports from various providers. Additionally spoofed emails can be simulated. To setup a complete mail infrastructure, several open source software packages are used. See table 3 for an overview of the used software.

Function	Software
Mail server	Postfix [15]
DMARC	OpenDMARC [16]
DKIM	OpenDKIM [17]
SPF	pypolicyd-spf [18]

Table 3: Software used for mail infrastructure during development and testing.

6.3 DMARC Pre-check

The first tool (see figure 2) is designed to give domain owners an overview of what potential impacts deploying DMARC will have. This tool can be used prior to deployment of DMARC, to estimate what part of actual email sent from the domain over a chosen measurement period would have profited from DMARC deployment.

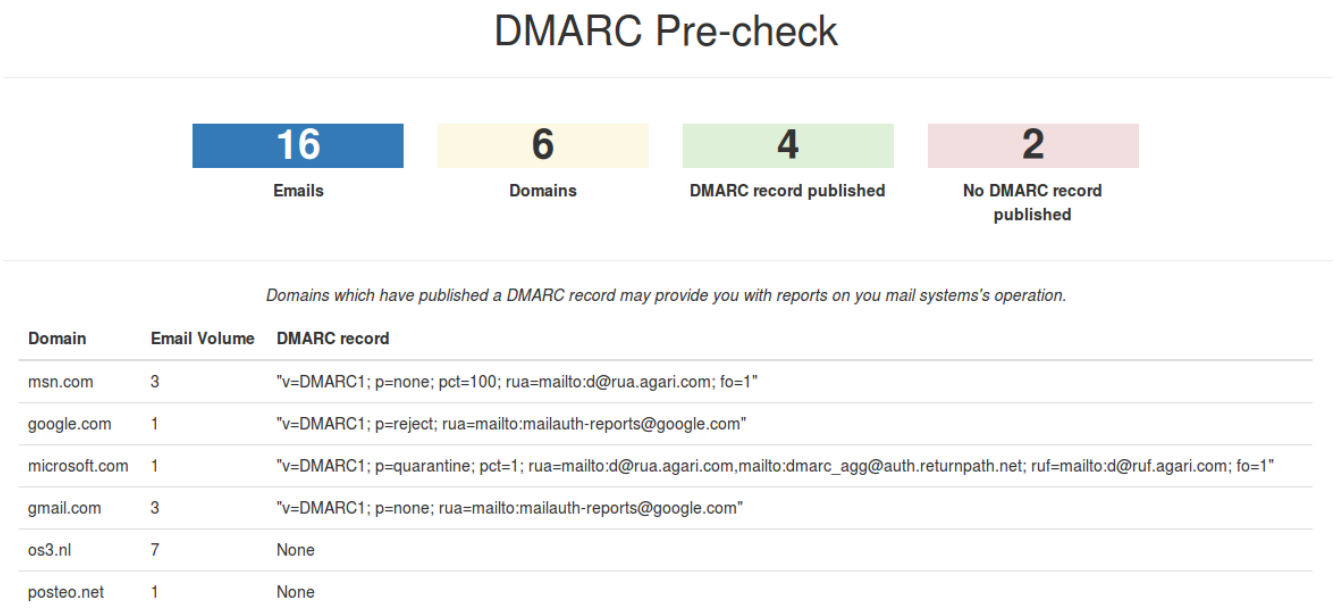

Figure 2: Web interface of tool 1

The tool starts with parsing the email logs to retrieve all domains to which email is sent. Next, it executes a DNS query for each found domain trying to retrieve a associated DMARC record. Together with the email volume the result is shown in a table. When deploying DMARC, the domain owner can expect to receive reports from domains that appear to have deployed DMARC. This assumption is made on the basis of the existence of a DMARC record. Not only can domain owners expect to receive reports from such domains, but also the enforcement of a published DMARC policy. Domains that are lacking a DMARC record and with which a relative high number of emails is exchanged, can be an incentive to actively encourage these domains in the deployment of DMARC.

6.4 DMARC monitor

The second tool is designed to guide the initial deployment of DMARC. Additionally, it can also be used to monitor the domain after initial deployment. This tool consists of several parts which are covered in the following subsection. For a complete overview, see figure 11.

6.4.1 DMARC status

The first widget found on web interface is the domain status box (see figure 3). This widget shows the current status of the DMARC DNS record by evaluating the presence of several important parameters. These include: Policy, Sub-policy, RUA, RUF and the percentage parameters. Besides the evaluation of key parameters, the complete DMARC record is also retrieved and shown. The goal of this widget is to inform the domain owner about the current DMARC configuration and provide warnings if any of the parameters are not set.

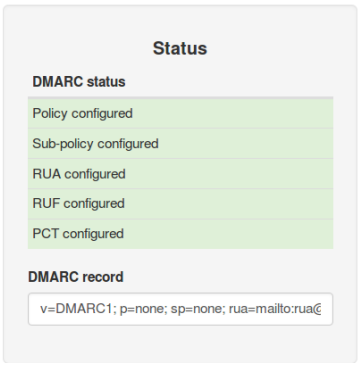

Figure 3: DMARC status widget

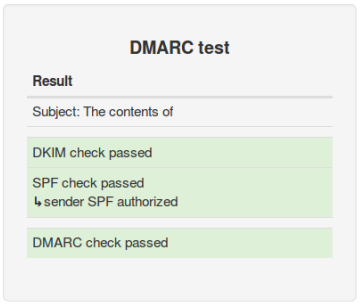

Figure 4: DMARC test widget

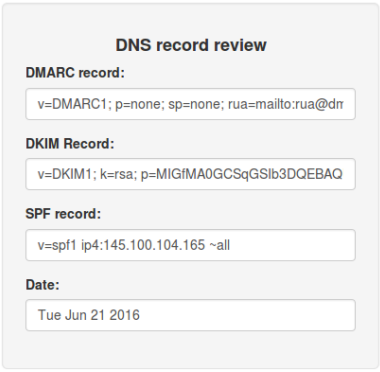

Figure 5: DNS record history widget

6.4.2 Automated DMARC test

Figure 4 shows the DMARC test widget. This widget is intended to test the configuration of SPF and DKIM . A separate remote mailbox must be reserved by the domain owner that will only be used for testing purposes. This mail box is used to simulate an end-to-end test. The application can automatically send an email to this reserved mailbox. For communication to the remote mailbox, a client server model is implemented. In this setup, the client is an application that reads the remote mailbox and the server is an application that will check emails on SPF and DKIM alignment which are sent from the client to the server. The server and the client communicate over a secure channel with each other. See figure 6 for visual representation. When a test message is sent to the reserved mailbox, the client application will connect to the server and send the received message. Consequently the server will perform SPF and DKIM checks, independent of software running on the remote mailbox or software running on the local infrastructure of the domain owner. If either DKIM or SPF pass, then the email is also consider to have passed DMARC . The DMARC alignment requirements (SPF or DKIM pass) can also be set to different values (e.g. both SPF or DKIM must pass before DMARC passes) . Each automatically sent email contains a time stamp that is present in the subject line. This time stamp is used to differentiate between tests.

6.4.3 DMARC authentication results

To give the user an overview of the aggregate results of incoming reports, a separate widget has been developed (see figure 7). The widget consists of three elements: an IP list, authentication results and a graph. This widget appears twice on the web interface, once for trusted sources and once for unknown sources (which possibly form a threat). Trusted sources are formed by a list of IP addresses that the user must supply. This list can be manually populated with sources or the user can make use of a developed tool that can automatically extract IP addresses from the SPF record of the domain. All IP addresses found in reports that are not contained within this list are marked as unknown and are possible threats. The IP list (leftmost block in figure 7) lists all IP address that fall within this category (once for trusted and once for unknown sources) together with the email volume originating from that address. This list is of most interest in the unknown/threat section. Addresses in this section might indeed be sources that try to spoof the monitored domain, however this list can also

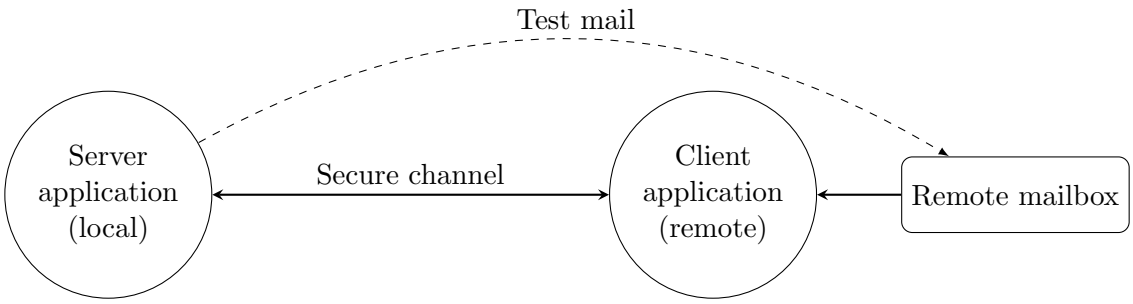

Figure 6: Figure depicting the workings of the automated DMARC tester.

contain legitimate sources. These sources can be misconfigured causing DMARC checks to fail. By monitoring this list, the user can track down possible misconfigured sources by comparing them to a list of legitimate sources. The second item contained within this widget are the authentication results. This item displays an aggregate summary of SPF, DKIM and DMARC authentication results. For this implementation, only a relaxed check is used (DMARC is considered passed when either SPF or DKIM passed). The strictness of the DMARC check is configurable (e.g. the DMARC check passes only if both the SPF and DKIM check pass). Together with the absolute volumes, the percentage (against the total volume) of results in each category is also shown. Finally, the right most item contains a graph that shows the DMARC authentication results. The green surface represents the number of messages that passed DMARC , while the red surface represents the volume of DMARC failed messages. The default setting for this graph is to show the authentication results of the past 30 days. Ideally the domain owner should not see any DMARC failures in the trusted section.

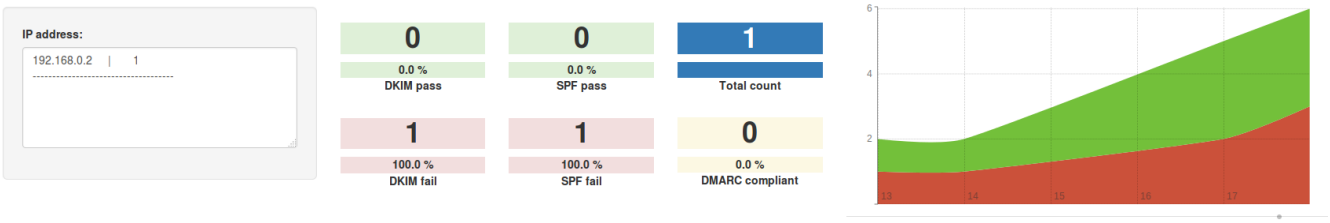

Figure 7: Widget showing aggregated DMARC authentication results.

6.4.4 DNS record history

Domain owners can make changes to their SPF, DKIM and DMARC records whilst in the deployment or during the operation of DMARC. These changes can affect authentication results and therefore a DNS record history widget is implemented. This widget records changes made to any of the mentioned DNS records and saves the new record in a SQL database. Underneath the graph in the previous widget (section 6.4.3), a time line will show points at which either of the three records have been altered. The user can review the record change in the history widget (see figure 5). The time line underneath the graph can help the domain owner to see what effects any change to the DNS records have on the authentication results of emails, both from trusted as well as untrusted (unknown/threats) sources.

6.5 Visualizations

6.5.1 AS visualization

Spoofed email can be send from a diverse set of sources. This diversity makes it hard to record and review the origin and quantity of spoofed email. An AS visualization is implemented to aid in the

tracking of email sources, the number of emails that is sent from an AS and the amount of emails that pass or fail SPF and DKIM checks. The visualization is implemented as a bubble chart that represents these three variables. The implementation categorizes each IP address found in the DMARC reports into an AS using a database to contains mappings between IP addresses and autonomous system. The implementation supports both IPv4 and IPv6 addresses. For each found AS (that is at least one email originated from this AS), the number of emails that passed either SPF or DKIM and the total number of emails is calculated. From these variables, a ratio is awarded to each AS. This ratio is calculated as:

$$Ratio = \frac{\text{Number of emails passed SPF or DKIM}}{\text{Total number of emails}} \tag{1}$$

The calculated ratio is represented by color with red being a low ratio and green being an high ratio. Legitimate sources should ideally have a ratio of 1.0. ASs that contain both legitimate sources as well as misconfigured sources will likely have ratios between 0.0 and 1.0. This can also be the case when the AS contains both legitimate as well as illegitimate sources. Lastly, sources that have a ratio 0.0 can be consider sources that try to spoof the domain. An exception are legitimate sources that are not configured properly.

An example of the bubble chart can be seen in figure 8. The numbers next to the bubbles represent a particular AS. The size of the bubbles depict the email volume originating from that AS and the color represents ratio calculated by formula 1. This visualization can help the domain owners to review malicious sources and their composition. Additionally, a text based reports is generated to supplement the graphical representation. This report contains the total email volume and the calculated ratio for each AS. Also, IP address from each AS are listed in the file together with authentication results. This enables the domain owner to monitor and possibly blacklist certain IP addresses. Furthermore, the user has an option to automatically retrieve the (organizational) name linked to the AS from a Regional Internet Registry.

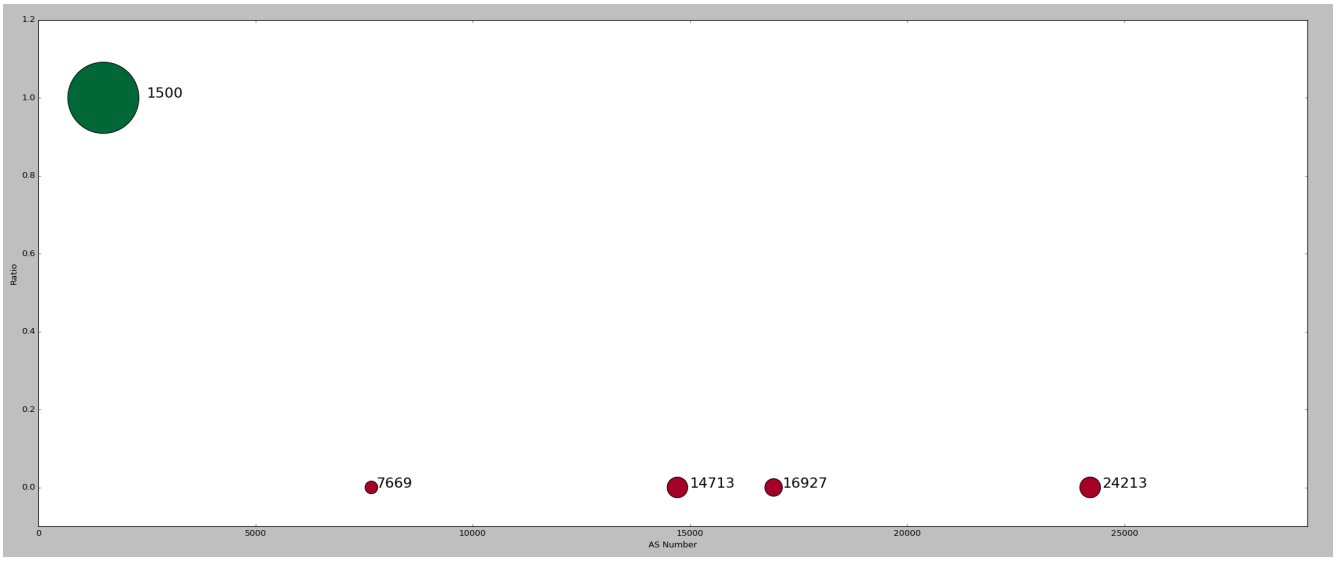

Figure 8: Example of a bubble chart visualization. In this case AS 1500 contains an legitimate source that is allowed to send email on behalf of the domain. The remaining autonomous system (7669, 14713, 16927, 24213) are likely to be sending illegitimate email which try to spoof the monitored domain as the ratio of these ASs is 0.0.

6.5.2 Heat map

Reports can originate from many different domains. To aid domain owners in reviewing the results from various sources, a heat map is implemented. An example can be seen in figure 12. The heat map consists of tiles that are color coded according to formula 1. A high ratio is resembled by a green tile whilst a low ratio is color coded red. Each tile represents one domain from which reports are received. Additionally, each tile shows the total number of messages received by a domain, the number of messages that passed DMARC and the number of messages that failed DMARC . Figures 9 and 10 show two example tiles (abbreviations used in the tiles: DMARC -P: DMARC pass, DMARC -F: DMARC failure).

Figure 9: Heat map tile with a high ratio.

Figure 10: Heat map tile with a low ratio

By examining the ratios, the domain owner can see which domains receive legitimate emails and those who do not. Domains with a relative low ratio can be subjected to a SPAM campaign. Domains can also fail in sending DMARC reports. To expose such domains, outgoing email can be monitored. After the email is sent, incoming reports are monitored to see if any feedback is received from the domain to which the email is sent. However experimentation shows that the domain to which email is sent, might not be the same domain from which reports are received. An example is the Google's Gmail service. When an email is sent to the *gmail.com* domain, DMARC reports will be sent from the *google.com* domain. This makes automatic checks difficult. One solution would be to manually map such domains.

A similar heat map is created for outgoing reports to give domain owners a grip on the diverse set of domains to which reports are sent. Also, one can review the authentication results per domain and see potential abuse of the domains. This implementation rests on the export functionality of OpenDMARC. This functionality allows one to exports email authentication results to an MySQL database. This database is used by OpenDMARC to generate aggregate DMARC reports which can be sent to domain owners.

7 Conclusion

Overall DMARC provides domain owners with various information about their domain. Firstly the domain owner can use DMARC feedback to monitor the deployment of DMARC . This is achieved by separating authentication results into trusted and unknown sources. Reviewing the the authentication results of trusted sources will reveal if SPF and DKIM are setup correctly. If any failures occur from trusted domains, the domain owner can retrieve the IP address of potentially misconfigured mail server from the DMARC reports.

Secondly, DMARC provides information to monitor illegitimate usage of a domain. This is achieved by reviewing authentication results and IP addresses present in the feedback reports. Sources that are not present in the trusted list and which fail SPF or DKIM can be considered illegitimate. These authentication results can be monitored (for example by graphing) to review illegitimate usage of the domain. IP addresses found in these reports can be cross checked against public SPAM databases to check if the source is known for sending illegitimate mail. If the IP source is not explicitly known to send SPAM and regular DMARC failures occur from this source, then the monitored domain may be subjected to a specific directed SPAM attack.

Thirdly, outgoing reports can also provide the domain owner with additional information about the monitored domain. Like incoming DMARC reports, outgoing reports contain authentication results of emails. In this case, reports contain the results of incoming messages. Incoming mails that fail the DMARC check can be potential sources of SPAM. By monitoring the authentication results, a possible SPAM attack against the monitored domain can be exposed. Moreover, by monitoring the outgoing reports, one can get a grip on the volume of reports that are sent to each domain and the corresponding authentication results. This can reveal the frequency of potential abuse per domain.

Although several commercial parties have created tools to parse and process DMARC data, several additional instruments are proposed in this research. Whilst the commercial tools focus on information extraction from incoming DMARC reports, additional sources like: outgoing reports, mail headers and DNS records measurements seem to be unused in these commercial tools.

The first proposed addition is a tool to monitor the outgoing reports as they can provide additional information as discussed earlier. Secondly an automatic DMARC testing tool is proposed. One of the key requirements in the correct setup of DMARC is the SPF and DKIM configuration. The proposed tool will check if an email sent from the monitored domain is DKIM and SPF aligned. This process can be executed completely autonomously, in contrast to the tool offered by one of the evaluated commercial tools. Additionally a DNS record history tool is proposed that records any changes made to the DKIM, SPF or DMARC DNS records. This can be used by the domain owner to correlate DNS record changes with DMARC authentication results of the email sent from legitimate and illegitimate sources. To visualize the diverse set of source, an AS visualization tool is proposed. Such a visualization tool should give the domain owner a concise overview of the sources that legitimately and illegitimately send email on behalf of the monitored domain. Furthermore, a SPAM checking tool is proposed which audits the IP address found in DMARC reports against public SPAM databases to access the reputation of the source which sent the message. Whilst commercial tooling aims at converting the reports into aggregated results and visualizations, one can have the needs to manually search through reports. Therefore a full text functionality on DMARC reports is proposed. Such search functionality should be build on top of a generic query language. Finally, a tool for checking DMARC endpoints is proposed. This tool aids the domain owner in finding domains from which no DMARC reports have been received. The use of commercial tooling bounds the user to redirecting reports to these commercial parties which might involve unwanted sharing of data. Therefore an open source implementation to monitor DMARC is built. This enables the domain to process reports within their own organization and offers the ability to customize the tools for specific needs. The implementation consist of two tools of which the first can be used to see what potential impact the deployment of DMARC will have. The second tool can be used to monitor the domain both when deploying DMARC as well as after deployment.

8 Future work

Further research can be done on the usage of information sources other than DMARC reports. This includes the email headers of messages that contain incoming DMARC reports.

Additionally, further research can be done on correlating information sources across different domains. One domain might encounter a source that is actively attempting to spoof the domain or sending SPAM messages. Parameters such as IP address(es) and email volume about this type of sources can be shared to review cross domain results and global trends.

Moreover, research can be done on an automatic deployment system. Such an system should advise the user what policy to apply based on different factors such as the number of emails that pass DMARC from trusted domains and the time spent in monitor mode.

References

[1] S. Kitterman. Sender policy framework (SPF) for authorizing use of domains in email, version 1. RFC 7208, RFC Editor, April 2014. http://www.rfc-editor.org/rfc/rfc7208.txt.

[2] D. Crocker, T. Hansen, and M. Kucherawy. Domainkeys identified mail (DKIM) signatures. STD 76, RFC Editor, September 2011. http://www.rfc-editor.org/rfc/rfc6376.txt.

[3] M. Kucherawy and E. Zwicky. Domain-based message authentication, reporting, and conformance (DMARC). RFC 7489, RFC Editor, March 2015. https://www.rfc-editor.org/rfc/rfc7489.txt.

[4] DMARC and the email authentication process. https://dmarc.org/overview/.

[5] H. Fontana. Authentication failure reporting using the abuse reporting format. RFC 6591, RFC Editor, April 2012. https://www.rfc-editor.org/rfc/rfc6591.txt.

[6] Ed M. Kucherawy. Domain-based message authentication, reporting and conformance (DMARC). Draft, March 2012. https://dmarc.org//draft-dmarc-base-00-02.txt.

[7] R. Danyliw, J. Meijer, and Y. Demchenko. The incident object description exchange format. RFC 5070, RFC Editor, December 2007. http://www.rfc-editor.org/rfc/rfc5070.txt.

[8] Spamhaus Project. Spamhaus Block List. https://www.spamhaus.org/pbl/.

[9] Dmarcian domain lifter. https://dmarcian.com/domain_lifter/.

[10] Dmarcanalyzer. https://www.dmarcanalyzer.com/.

[11] A free tool to monitor and implement DMARC. https://dmarc.postmarkapp.com/.

[12] Core team Mark Otto, Jacob. Bootstrap. http://getbootstrap.com/.

[13] Douglas Hunter David Chester and Silas Sewell at Shutterstock. Rickshaw. http://code.shutterstock.com/rickshaw/.

[14] TechSneeze. dmarcts-report-parser. https://github.com/techsneeze/dmarcts-report-parser/blob/master/dmarcts-report-parser.pl.

[15] Wietse Venema and other contributors. Postfix. https://launchpad.net/pypolicyd-spf/.

[16] The Trusted Domain Project. OpenDMARC. http://www.trusteddomain.org/opendmarc/.

[17] The Trusted Domain Project. OpenDKIM. http://www.opendkim.org/.

[18] Scott Kitterman. pypolicyd-spf. https://launchpad.net/pypolicyd-spf.

Appendices

The source code of the developed tooling can be found at: `https://github.com/cheatas/DMARCate`.

Figure 11: Web interface of the DMARC monitor.

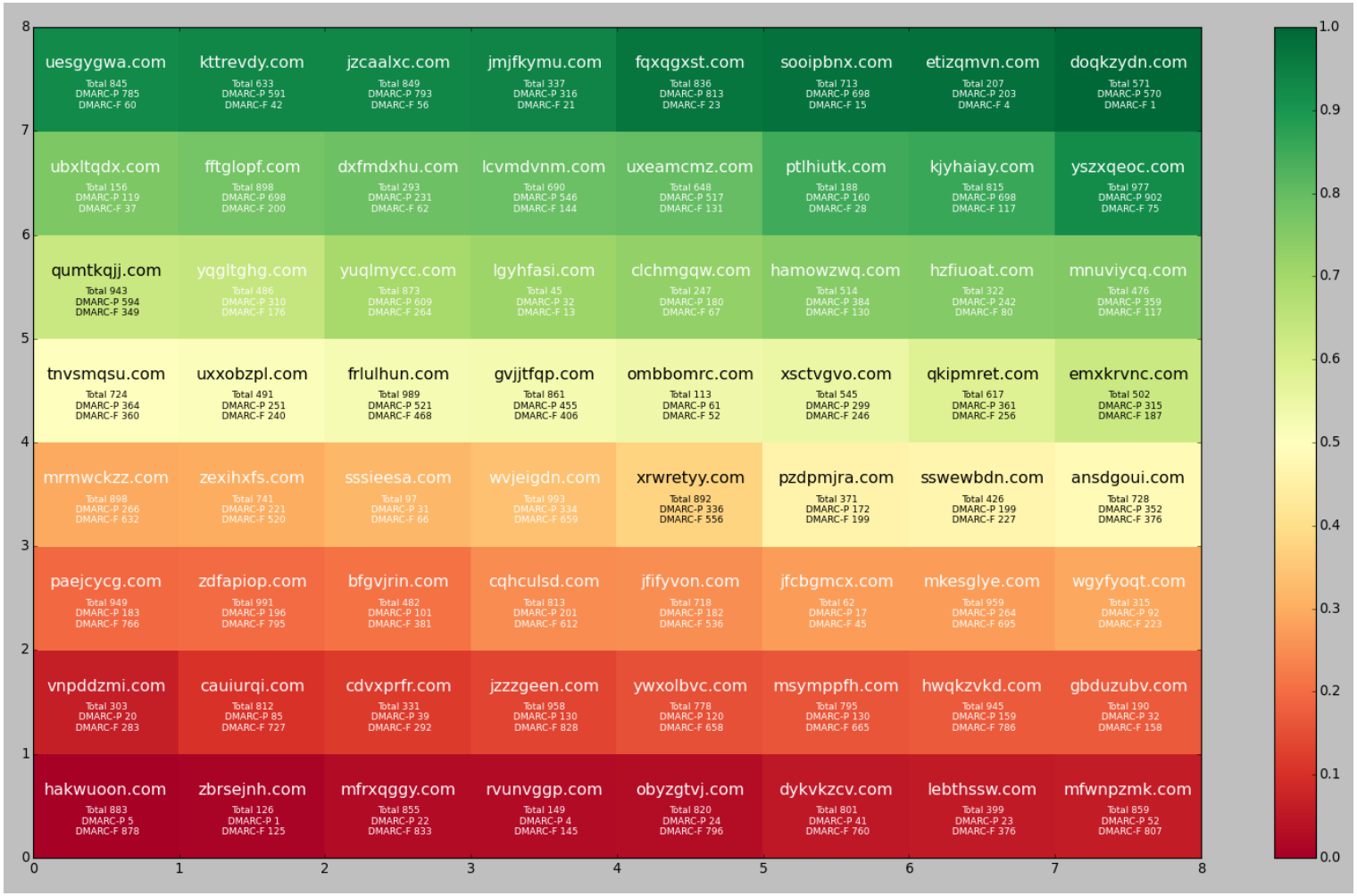

Figure 12: Heatmap showing authentication result from different domains.

```
Message-ID: <433689.81121.example@mta.mail.receiver.example>
From: "SomeISP Antispam Feedback" <feedback@mail.receiver.example>
To: arf-failure@sender.example
Subject: FW: You have a new bill from your bank
Date: Sat, 8 Oct 2011 15:15:59 -0500 (CDT)
MIME-Version: 1.0
Content-Type: multipart/report;
boundary="------------Boundary-00=_3BCR4Y7kX93yP9uUPRhg";
report-type=feedback-report
Content-Transfer-Encoding: 7bit

--------------Boundary-00=_3BCR4Y7kX93yP9uUPRhg
Content-Type: text/plain; charset="us-ascii"
Content-Disposition: inline
Content-Transfer-Encoding: 7bit

This is an authentication failure report for an email message
received from a.sender.example on 8 Oct 2011 20:15:58 +0000 (GMT).
For more information about this format, please see [RFC6591].

--------------Boundary-00=_3BCR4Y7kX93yP9uUPRhg
Content-Type: message/feedback-report
Content-Transfer-Encoding: 7bit

Feedback-Type: auth-failure
User-Agent: Someisp!Mail-Feedback/1.0
Version: 1
Original-Mail-From: anexample.reply@a.sender.example
Original-Envelope-Id: o3F52gxO029144
Authentication-Results: mta1011.mail.tp2.receiver.example;
dkim=fail (bodyhash) header.d=sender.example
Auth-Failure: bodyhash

DKIM-Canonicalized-Body: VGhpcyBpcyBhIG1lc3NhZ2UgYm9keSB0
aGF0IGdvdCBtb2RpZmllZCBpbiB0cmFuc2l0LgoKQXQgdGhlIHNhbWU
gdGltZSB0aGF0IHRoZSBib2R5aGFzaCBmYWlscyB0byB2ZXJpZnksIH
RoZQptZXNzYWdlIGNvbnRlbnQgaXMgY2xlYXJseSBhYnVzaXZlIG9yI
HBoaXNoeSwgYXMgdGhlClN1YmplY3QgYWxyZWFkeSBoaW50cy4gIElu
ZGVlZCwgdGhpcyBib2R5IGFsc28gY29udGFpbnMKdGhlIGZvbGxvd2l
uZyB0OZXh0OgoKICAgUGxlYXNlIGVudGVyIHlvdXIgZnVsbCBiYW5rIG
NyZWRlbnRpYWxzIGF0CiAgIGh0dHA6Ly93d3cuc2VuZGVyLmV4YW1wb
GUvCgpXZSBhcmUgaaW1wbHlpbmcgdGhhdCwgYWx0aG91Z2ggbXVsdGlw
bGUgZmFpbHVyZXMKcmVxdWlyZSBtdWx0aXBsZSByZXBvcnRzLCBhIHN
pbmdsZSBmYWlsdXJlIGNhbiBiZQpyZXBvcnRlZCBhbG9uZyB3aXRoIH
BoaXNoaW5nIGluIGEgc2luZ2xlIHJlcG9ydC4K
DKIM-Domain: sender.example
DKIM-Identity: @sender.example
DKIM-Selector: testkey
Arrival-Date: 8 Oct 2011 20:15:58 +0000 (GMT)
Source-IP: 192.0.2.1
```

```
51 Reported-Domain: a.sender.example
52 Reported-URI: http://www.sender.example/
53
54 --------------Boundary-00=_3BCR4Y7kX93yP9uUPRhg
55 Content-Type: text/rfc822-headers
56 Content-Transfer-Encoding: 7bit
57
58 Authentication-Results: mta1011.mail.tp2.receiver.example;
59 dkim=fail (bodyhash) header.d=sender.example;
60 spf=pass smtp.mailfrom=anexample.reply@a.sender.example
61 Received: from smtp-out.sender.example
62 by mta1011.mail.tp2.receiver.example
63 with SMTP id oB85W8xV000169;
64 Sat, 08 Oct 2011 13:15:58 -0700 (PDT)
65 DKIM-Signature: v=1; c=relaxed/simple; a=rsa-sha256;
66 s=testkey; d=sender.example; h=From:To:Subject:Date;
67 bh=2jUSOH9NhtVGCQWNr9BrIAPreKQjO6Sn7XIkfJVOzv8=;
68 b=AuUoFEfDxTDkHlLXSZEpZj79LICEps6eda7W3deTVFOk4yAUoqOB
69 4nujc7YopdG5dWLSdNg6xNAZpOPr+kHxt1IrE+NahM6L/LbvaHut
70 KVdkLLkpVaVVQPzeRDI009SO2Il5Lu7rDNH6mZckBdrIx0orEtZV
71 4bmp/YzhwvcubU4=
72 Received: from mail.sender.example
73 by smtp-out.sender.example
74 with SMTP id o3F52gxO029144;
75 Sat, 08 Oct 2011 13:15:31 -0700 (PDT)
76 Received: from internal-client-001.sender.example
77 by mail.sender.example
78 with SMTP id o3F3BwdY028431;
79 Sat, 08 Oct 2011 13:15:24 -0700 (PDT)
80 Date: Sat, 8 Oct 2011 16:15:24 -0400 (EDT)
81 Reply-To: anexample.reply@a.sender.example
82
83 From: anexample@a.sender.example
84 To: someuser@receiver.example
85 Subject: You have a new bill from your bank
86 Message-ID: <87913910.1318094604546@out.sender.example>
87
88 --------------Boundary-00=_3BCR4Y7kX93yP9uUPRhg--
```

Listing 3: Example forensics reports from RFC 6591.[5] Lines 83 till 86 show the original headers of the mail that was received by generator of the report. It includes the From, To, Subject and orginal message-ID. Additional information includes results of authentication test performed at the report generator and date/time information.