Spoofed Emails: An Analysis of the Issues Hindering a Larger Deployment of DMARC

Olivier Hureau[1], Jan Bayer[12], Andrzej Duda[1], and Maciej Korczyński[1]

[1] Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, Grenoble, France
{firstname.lastname}@univ-grenoble-alpes.fr
[2] KOR Labs Cybersecurity, France
{firstname.lastname}@korlabs.io

Abstract. In 2015, the IETF released an informational specification for the DMARC protocol, not establishing it as an Internet standard. DMARC is designed to fight against email spoofing, on top of SPF and DKIM. Given that these anti-spoofing measures could lead to the loss of legitimate emails, DMARC embedded a reporting system enabling domain owners to monitor rejected messages and enhance their configurations. Research communities have extensively examined various aspects of DMARC, including adoption rates, misuse, and integration into early spam detection systems while overlooking other vital aspects, potentially impeding its broader use and adoption.
This paper sheds light on a widespread lack of comprehension of the standard and unexpected behavior regarding DMARC among various groups, including professionals, open-source libraries, and domain owners. We propose measurement and analysis approaches that include a DMARC record parser, a methodology for dataset collection, and an analysis of the domain name landscape. We provide insights for fostering a deeper understanding of the DMARC ecosystem.
We also identify email addresses in DMARC records belonging to 9,121 unregistered domain names, which unintended users could register, leading to potential data leakage from the email systems of domain owners.

Keywords: Email anti-spoofing mechanisms, DMARC, SPF, DKIM

1 Introduction

In the current email distribution system based on the Simple Mail Transfer Protocol (SMTP) [25], it is relatively easy to spoof messages: a malicious actor just sends a message with a forged sender address and other parts of the email header to appear as sent from a legitimate source.

Internet Engineering Task Force (IETF) specified several email anti-spoofing schemes in *security extensions* such as the Sender Policy Framework (SPF) [24], the DomainKeys Identified Mail (DKIM) [7], and Domain-based Message Authentication, Reporting, and Conformance (DMARC) [28]. They aim at authenticating the sender and deciding what to do with suspicious emails. The extensions define a set of rules that specify the servers allowed to send emails on behalf of a domain name and provide strategies for dealing with spoofed messages. If properly configured, the anti-spoofing mechanisms allow the recipient of an email to verify that the sender domain name is legitimate.

However, some legitimate emails may get rejected because of misconfigured or too tight anti-spoofing mechanisms. Thus, domain administrators must precisely set up the SPF/DKIM parameters of their domains to avoid the loss of legitimate emails. Although the email receiver can apply their own policies and actions regarding the SPF and DKIM results, a domain owner, through her DMARC record, can provide the expected behavior the email receiver should undertake when receiving a message failing the DMARC check mechanism.

Several studies considered the operation of the anti-spoofing mechanisms via active and passive measurements [18,13,44,21,39,34,10,49,48,51,36,2,8,46]. Much effort focused on active scans and the analysis of DMARC deployment across popular domain names [18,13,44,21,46,51,8] as well as for the overall population of domain names [39,34,2,36]. The studies concluded that the adoption of DMARC is still low and subject to misconfigurations and vulnerabilities [3,5,35,2].

In this paper, we present a large-scale study of DMARC to observe the user habits and preferences, consider the evolution of DMARC adoption in time, and understand how popular domains use DMARC. Our measurements indicate that DMARC is frequently not well understood or effectively used. There are several reasons for this state of affairs—we identify four main problems:

- Specifications are complex, occasionally ambiguous, and at times contradictory, with a multitude of over thirty RFCs interlinked with intricate dependencies in the realm of anti-spoofing mechanisms. Some of these RFCs have been abandoned, updated, or rendered obsolete, potentially resulting in diminished understanding, suboptimal configurations, or possibly misapplications.
- Although DMARC checker tools are designed to help users create and configure their DMARC records, they can generate false positives and false negatives, potentially resulting in inaccurate evaluations of the records' validity and effectiveness.
- Progressive improvement of configurations is tedious due to a suboptimal, at times incorrect, or delegated use of DMARC reporting.
- Some domain owners may choose not to adopt DMARC, either due to a perceived lack of added value or skepticism about its effectiveness. For cases with limited benefits of using DMARC, they might not allocate resources to its deployment.

In summary, the paper brings the following contributions:

- We propose a methodology for gathering DMARC-related data: parsing DMARC records, analyzing protective means used by domain owners and the prevalence of various DMARC tags, URIs specified in `rua` or `ruf` tags, and collecting statistics on popular domain names. We also report on the

time evolution of DMARC policies. Our analysis suggests that DMARC is not well understood by domain owners.
- We gather statistics on DMARC report receivers to identify the main stakeholders involved in report processing: we show that three the most important third-party services (Proofpoint, Mailinblue, and Agari) represent 21% of those present in DMARC records.
- We discover a vulnerability related to email addresses in DMARC records that may allow attackers to retrieve DMARC reports.
- We assess the compliance of online DMARC checkers and open source libraries with RFC 7489 and observe that none of them fully comply with the standard. To improve this situation, we have developed a Python-based DMARC parser based on the Augmented Backus-Naur Form (ABNF) that adheres to the syntactic rules of RFC 7489 and RFC 6376, to be shared with the community.
- We analyze the collected statistics and formulate recommendations aiming at simplifying the DMARC specifications and making them more clear to enable their larger adoption and deployment.

2 Related Work

Over the years, IETF strived to enhance email security by proposing, refining, and updating SPF, DKIM, and DMARC anti-spoofing mechanisms with many RFCs. The protocols have already demonstrated their effectiveness as a means for securing the email distribution system [19,32,45]. However, previous work also revealed vulnerabilities in their implementation [3,5], and explored possible misuse [35,2]. SPF, DKIM, and DMARC records and email reception logs have been used to study other vulnerabilities [41], or they were integrated into early spam detection systems [9,15,43,26].

Previous research extensively investigated their adoption through both active and passive measurements [18,13,44,21,39,34,10,49,48,51,36,2,8], with a particular focus on analyzing DMARC deployment across popular domain names [18,13,44,21,46,51,8] and the broader population of domain names [39,33,34,2,36]. Only Czybik et al. [8] indicated which software and methodology they used to parse DMARC records.

Hu et al. [20] aimed at understanding the reasons behind their limited adoption. They concluded that significant effort is needed to address technical issues and create incentives for widespread adoption within the community. The studies by Portier et al. [39] and Ishtiaq et al. [2] are the only ones that present the statistics regarding the prevalence of `rua` and `ruf` tags in DMARC records.

Our analysis involves inspecting the domain name part of the email addresses specified in the DMARC record (`rua` and `ruf` tags) to identify domains available for registration that can be set up by attackers to receive DMARC reports. Moreover, our results are consistent with prior studies demonstrating how misspelled or expired domains can compromise the security of both users and systems [47,42,30,29,13,31].

We propose measurement and analysis approaches that include a DMARC record parser and a methodology for dataset collection and analysis. Our findings highlight the lack of understanding among various stakeholders and software, offering valuable insights for its improvement.

3 Background

In this section, we provide an introduction to the email ecosystem followed by an overview of three key mechanisms that help ensuring email integrity and prevent domain name spoofing: SPF, DKIM, and DMARC. In this context, we discuss DMARC reporting, the mechanism that provides administrators with information on email activity related to their domains including SPF, DKIM, and DMARC authentication checks.

3.1 Simple Mail Transfer Protocol (SMTP)

SMTP is a protocol for sending and receiving email messages, specified first in 1982 by RFC 821 [40] and further refined by the current standard RFC 5321 [25]. Despite being widely used, SMTP is inherently insecure because it lacks built-in mechanisms for authentication and encryption, making it vulnerable to eavesdropping, domain name spoofing, and other forms of email abuse. As a result, modern email systems often use additional security protocols such as Transport Layer Security (TLS) and anti-spoofing mechanisms: SPF, DKIM, and DMARC to mitigate its design flaws.

Emails are sent using a Mail Transfer Agent (MTA) from the sender to the recipient MTA. Then, the Mail Delivery Agent (MDA), referred to as the receiver, queries the name server of the sender domain to check the SPF, DKIM, and DMARC records of the sender domain. If the checks are successful, the email is delivered to the recipient inbox.

3.2 Sender Policy Framework (SPF)

RFC 4408 [50] defined SPF as an experimental protocol in 2006 and it was further refined in RFC 7208 [24]. The purpose of SPF is to enable an email receiver to identify the hosts authorized to send emails on behalf of a domain name based on the information published in the DNS `TXT` Resource Records (RRs) of the domain (called SPF records). An SPF record needs to start with the version string `v=spf1` and provides the specification of the authorized email senders for the domain by the following SPF mechanisms: `a`, `ip4`, `mx`, `all`, `include`, `exists`, `redirect`. For instance, if there is the `A` record `example.com A 198.51.100.1` in DNS for the domain `example.com`, the following SPF record `v=spf1 a ip4:192.0.2.0/24 -all` indicates that only hosts with the IP address of `198.51.100.0` (the `a` mechanism), or with the IP address in the `192.0.2.0/24` prefix (the `ip4` mechanism) are permitted senders, all others are forbidden (the `-all` mechanism).

Upon the reception of an email, the mail receiver executes the `check_host()` function on the domain name specified in the `Mail From` address [24] that checks the SPF record for the domain to determine whether the host sending the email is authorized. The validation result can be `neutral`, `pass`, `fail`, `soft fail`, `temperror`, or `permerror`. If the result is `fail`, `permerror`, or `temperror`, the mail receiver may reject the email, depending on its anti-spoofing procedures.

3.3 DomainKeys Identified Mail (DKIM)

RFC 4871 [1] defined DKIM in 2007 and it was obsoleted by RFC 6376 in 2011 [7]. DKIM specifies the authentication and integrity verification of email messages using public-key cryptography according to the principles stated in RFC 4870 [11]. The sender of an email uses its private key to generate a digital signature for the email, adds a header that includes a hash of the signature and the selector of the associated public key. The `TXT` record of `<selector>._domainkey.example.com` contains the public key used for the signature. The mail receiver can verify the digital signature, which gives one of the following results: `success`, `permfail`, or `tempfail`. An email can contain multiple DKIM signatures. If at least one of them is valid, the evaluation is successful.

3.4 Domain-based Message Authentication, Reporting, and Conformance (DMARC)

DMARC [28] builds on top of SPF and DKIM to specify how mail receivers should treat the emails that fail authentication checks.

For a given domain name, a `TXT` RR stored in the `_dmarc` subdomain (called a DMARC record) specifies the DMARC handling policy. When an email receiver receives a message, it performs the DMARC check. If the check fails, the handling policy specifies the actions that the email receiver should undertake. DMARC also provides reporting capabilities that allow domain owners to receive feedback on how their emails are treated. In the following, we review the DMARC format and its most common rules.

DMARC Check Mechanism. DMARC associates the names verified by SPF and DKIM with the content of the `FROM:` field in the email header (referred to as the `Author Domain` [28]). This association is established through the concept of *alignment*, meaning that these domain names must match (or partially match in the case of a relaxed configuration). The evaluation results in a 'success' for DKIM and a 'pass' for SPF. Both the DKIM evaluation and the SPF `check_host()` functions are executed on the `Author Domain`. An email is deemed to satisfy the DMARC check mechanism if either SPF **or** DKIM are *aligned*. The DMARC check mechanism fails if and only if both SPF **and** DKIM are not aligned (this conjunction is usually not well understood).

Figure 1 provides an overview of the DMARC check mechanism involving Alice and Mallory sending an email to the Bob's email address: `bob@example.com`

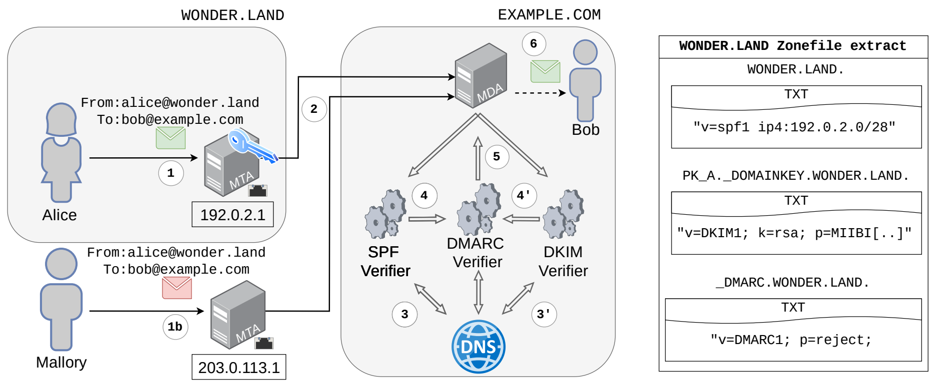

Fig. 1: DMARC check mechanism overview

(①, 1b). In this scenario, Alice is a legitimate user of `wonder.land`,[3] while Mallory attempts to spoof the Alice's email address, `alice@wonder.land`. Both Mallory's and Alice's MTAs connect to the Bob's `example.com` MDA and transfer the email starting with the command `MAIL FROM: wonder.land` (②). The MDA runs the SPF `check_host()` function on the `wonder.land` domain name. Since the SPF records for `wonder.land`, as retrieved by the Bob's MDA (④), specify permission for the `192.0.2.0/28` IPv4 range, the Alice's MDA SPF check is successfully passed because `192.0.2.1` is designated as a permitted sender. In this context, the Alice's SPF is considered *aligned*, while the Mallory's SPF is not aligned.

In the Alice's email, there is a `DKIM-Signature` with the `pk_a` selector. The Bob's MDA retrieves the `TXT` records at `pk_a._domainkey.wonder.land` (③'). The signature in the email matches the public key in the DKIM records. The result of the DKIM check is `success`. In this scenario, the Alice's DKIM is considered *aligned*.

The Bob's MDA retrieves the `wonder.land` DMARC record at `_dmarc.wonder.land`. It specifies the `p=reject` handling policy (⑤). Since the Alice's email DKIM or SPF are aligned, the DMARC check (⑤) passes, and the email is successfully delivered to Bob (⑥). However, in the case of Mallory, whose DKIM and SPF are not aligned, the DMARC check fails. According to the `wonder.land` DMARC record, the domain owner wants the rejection of the Mallory's email.

DMARC Record Format. A valid DMARC record must start with `v=DMARC1` and be unique. Domain name owners may specify multiple policies represented as tags separated by semicolons (the `p` tag is mandatory, other tags are optional, and some of them have default values). Any tag that does not conform to RFC must be ignored. The tags are defined as follows:

– `p`: requested handling policy with three possible options:

[3] A fictional domain name.

reject, quarantine, or none. When an email fails the DMARC check, indicating that both SPF and DKIM are not aligned, the email receiver is expected to take one of the following actions based on the value of the p tag: reject the email if p=reject, flag the email as suspicious (e.g., by directing it to a quarantine or spam folder) if p=quarantine, or take no action if p=none.

- sp: requested handling policy for subdomains. The options for sp are the same as for p, and by default, sp takes the same value as p. For instance, when the domain owner of example.com specifies p=none and sp=reject, she requests that the email failing the DMARC check with the example.com Author Domain should be accepted. However, any email with a subdomain of example.com (e.g., email.example.com) as the Author Domain should be rejected.
- adkim and aspf: DKIM, and SPF alignment modes with the following values: s meaning strict and r relaxed (default). In strict mode, the authenticated domain and the Author Domain must be the same. The relaxed mode accepts that both names are in the same organizational domain. For instance, if the policy is aspf=r, and if an email with the Author Domain example.com is sent from the host email.example.com passing the SPF checks for email.example.com, the email will be aligned because example.com and email.example.com are within the same organizational domain.
- rua and ruf: specify one or several URIs (e.g., an email address) for receiving the aggregate (rua) and failure (also called forensic) (ruf) reports. While email receivers are expected to send reports, it is not an obligation (as per RFC 2119, which uses SHOULD to indicate a recommended action [4]). Nonetheless, these reports can provide valuable insights into email management and serve as a monitoring instrument for uncovering domain name abuse.
- ri: aggregate reporting interval. By default set to 86400 s, aggregate reports are generated on a daily basis. However, a domain name owner can specify the time frame for which she wants to receive aggregated reports.
- fo: failure reporting options. By default '0', it indicates which anti-spoofing mechanism triggers the event of sending a failure report to the URIs specified in the ruf tag:
  - 0: generate a DMARC failure report if SPF *and* DKIM are not aligned (default option),
  - 1: generate a DMARC failure report if SPF *or* DKIM are not aligned,
  - d: generate a DKIM failure report if DKIM is not aligned,
  - s: generate an SPF failure report if SPF is not aligned.

  Note that the requested handling policy is not affected by the fo tag.
- pct: sampling rate. The pct tag accepts an integer between 0 and 100 (default) that indicates the percentage of emails subject to the DMARC handling policy. However, it does not have any impact on the reporting system. The DMARC check procedure is still executed, and the outcome of the check is reported [28].

DMARC Feedback. While both SPF and DKIM have their own reporting mechanisms defined in their respective RFCs [23,27], DMARC is the primary email authentication protocol that leverages both SPF and DKIM to provide a unified and aggregated reporting mechanism and has become a *de facto* industry standard for reporting on email processing. We provide more information on aggregate and failure reports below.

- *Aggregate reports* : an aggregate report contains statistical data on the authentication results of emails received by DMARC-compliant mail receivers during a specific period, usually 24 hours. The data includes both emails that passed the DMARC check and those that failed. The report helps domain owners to monitor and evaluate the effectiveness of their DMARC policy, identify issues with their email authentication setup, and stop any unauthorized use of their domain for malicious purposes. Aggregate reports are generated and automatically sent to the email address specified in the `rua` tag of the DMARC record for the domain.
- *Failure Reports*: a failure report is a feedback mechanism that provides information about email messages that failed SPF and/or DKIM checks. The report is sent to the email address specified in the `ruf` tag of the DMARC record for the domain according to the failure reporting options (`fo` tag). It includes detailed information about the failed message, such as the message headers and the reasons for the failure. The failure report may include the email that did not pass the authentication mechanism as an attachment. The purpose of failure reports is to help domain owners identify and stop any unauthorized use of their domain for malicious purposes or determine any misconfiguration. Failure reports are in a standard, machine-readable format called ARF (Abuse Reporting Format) defined by RFC 6591 [17].

External Destination Verification. A potential vulnerability exists regarding URIs specified in `rua` and `ruf` tags. An attacker can specify the email address of their victim in the `rua` or `ruf` tags and cause email receivers to send reports. This may result in unsolicited emails flooding the mailbox of the victim. By design, DMARC is immune to such a scenario because the email receiver is requested to perform an *external destination verification*. Let us assume that an external domain name in the `rua` or `ruf` tag of a given monitored domain name is not within the same organizational domain. The external destination verification involves checking if the domain name has a specific `TXT` record that can be queried at the domain name formed by appending the monitored domain name, the string `._report._dmarc.`, and the external domain name.

```
v=DMARC1;p=none;sp=reject;fo=1:d;ruf=mailto:ruf@security.example.com;
rua=mailto:rua@example.com,mailto:dmarc@help.example.org;ri=43200
```

Fig. 2: Example of DMARC record found for the `example.com` domain name

As an example, the record at figure 2 contains three email addresses: `ruf@security.example.com`, `rua@example.com`, and `drua@example.com`. Given that `dmarc-ag@example.com` and `ruf@security.example.com` are part of the `example.com` organizational domain name, the external destination verification are not performed for these URI. However, an email receiver should retrieve the DNS `TXT` record of `example.com._report._dmarc.help.example.org`. Given the result `"v=DMARC1;"`, the domain owner of `help.example.org` permits email receiver to send the DMARC report towards `example.com` and from any email address `'@help.example.org'`.

4 DMARC Large Scale Measurements

In this section, we begin with an overview of our measurement platform and the raw data obtained from DNS queries. Then, we delve into the protective measures selected by domain owners and analyze the data regarding the prevalence of various DMARC tags to gain insights into different behaviors. Subsequently, we present statistics related to URIs specified in `rua` or `ruf` tags. Finally, we focus on popular domain names.

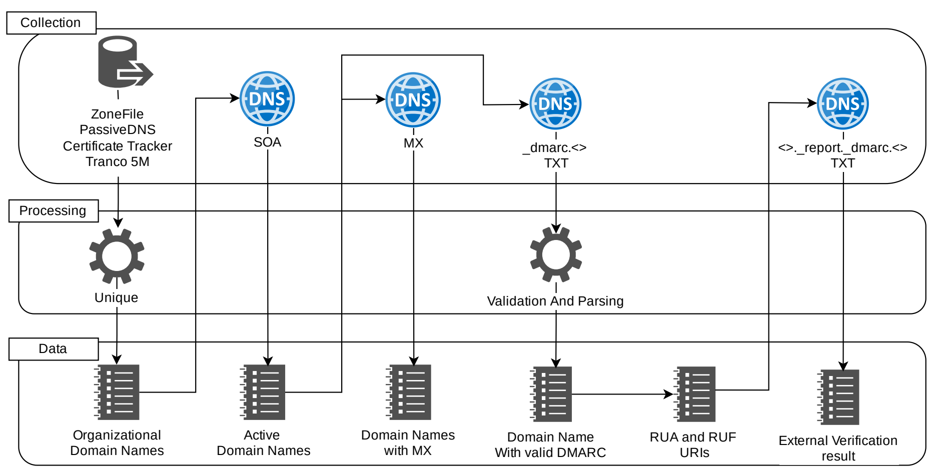

Fig. 3: Overview of the methodology for collecting DMARC data

We have used `zdns` [22] to conduct a large-scale data collection campaign to analyze the feedback information expected from DMARC reports.

First, we have created a list of domain names by collecting data from various feeds: the generic Top-Level Domain (gTLD) zone files from the ICANN Centralized Zone Data Service (CZDS),[4] passive DNS data from the SIE Europe,[5]

[4] `https://czds.icann.org`
[5] `https://www.sie-europe.net`

domain names from the Google certificate transparency logs,[6] and the Tranco 5 M list[7] [38].

We have used the Mozilla public suffix list[8] to extract only the *organizational domain names*. The aggregation of the domain names extracted from our feeds results in 513 M unique domain names. Since the list contains unregistered domain names, we have queried the `SOA` (Start of Authority) RR for each domain and excluded those with an `NXDOMAIN` response.

We then have queried the `TXT` records for the subdomain `_dmarc` to obtain DMARC records [28] and parse them using our parser described later. As they may include an external address in the `rua` or `ruf` tags, we have carried out the external destination verification of the addresses.

We scanned the `SOA`, `TXT` (DMARC) records, and performed the external destination verification both in September 2022 and October 2023.

Table 1: DNS scan results

	Dataset description				Measurements Numbers				
	Name	Record Type	Domain Prefix	Dataset Size	Noerror	Nxdomain	Other	Empty Records	Successfully Retrieved
09/2022	M_1	SOA	$\emptyset$	287.9 M	273 M	12.6 M	2.1 M	15.2 M	257.7 M
	M_2	MX	$\emptyset$	257.7 M	257.3 M	155 K	317 K	145.2 M	112.1 M
	M_3	TXT	_dmarc.	257.7 M	87.6 M	168.6 M	1.5 M	65.4 M	11.6 M
	M_4	TXT	<>._report._dmarc.	7,9 M	4,4 M	3,4 M	23 K	205 K	4,2 M
10/2023	M_5	SOA	$\emptyset$	513.7 M	286.5 M	212.0 M	15.1 M	6.2 M	280.3 M
	M_6	MX	$\emptyset$	280.3 M	273 M	12.7 M	2,145 K	119.4 M	153.6 M
	M_7	TXT	_dmarc.	280.3 M	95.5 M	182.4 M	2.3 M	68.8 M	15.9 M
	M_8	TXT	<>._report._dmarc.	5.4 M	4.26 M	1.1 M	13 K	126 K	4.13 M

Table 1 shows the results of active DNS measurements related to DMARC. We get the status and the content of each DNS response and parse the records to keep only the valid ones. For each measurement denoted by $M_{[1-8]}$, we provide the requested RR type, the prefix of the domain, the size of the collected dataset, the DNS status and the numbers of each status type, the number of empty records, and the number of valid (according to the related RFC) records after parsing it.

A DNS query returns different DNS error codes: i) `NOERROR` when the query was successful, ii) `NXDOMAIN` for the domain name not present in the DNS zone file of the queried name server, and iii) OTHER for all remaining error codes such as `TIMEOUT`, `SERVFAIL`, or `REFUSED`.

Even if the returned DNS error code is `NOERROR`, it does not imply that the answer contains any Resource Record. The "Empty records" column in Table 1 corresponds to the answers with `NOERROR` and no data inside. The "Successfully

[6] https://googlechrome.github.io/CertificateTransparency

[7] https://tranco-list.eu

[8] https://publicsuffix.org/

retrieved" column contains the number of domain names for which we have obtained valid data in response to a given query. For instance, when looking for the DMARC record in M_3, we query the `TXT` record of the `_dmarc` subdomain and parse it to validate its content.

To begin our scans, we have collected and parsed the `SOA` records for the 513 M (M_1) domains from our aggregated list to exclude unsuitable domain names. We kept only domain names with the `SOA` records with status `NOERROR` and excluded those with empty records. As a result, our dataset contains 280.3 M (286.5 M - 6.2 M) domain names. This dataset is used to perform measurements M_6 and M_7.

Measurements M_7 involve querying the `TXT` record for the domain name with the prefix `'_dmarc.'`. The DNS answer, the `RRset`, may contain multiple records. We have then parsed all RRs and excluded the invalid strings: either because the content was invalid [9] or because the domain had more than one valid record (a record is a valid DMARC record if only one RR is syntactically correct). Around 150 K domain names had an RR containing the 'dmarc' string but were not syntactically correct, and 68 K contained multiple valid RRs. Thus, for the 16.7 M (95.5 M - 68.8 M) domain with a non-empty answer, the RRset contained in total 295 M RRs. Only 15.9 M domain names had a valid DMARC record.

Measurements M_8 is the External Destination Verification. We have found 3.3 M domain names and a total of 5.4 M email addresses for which the External Destination Verification should be processed. 20% of these requests result in either `NXDOMAIN`, `SERVFAIL`, `REFUSED` or `TIMEOUT`. Finally, 4.1 M (75%) verifications succeeded.

4.1 DMARC as a Domain Name Protection Mechanism

Table 2: DMARC handling policies according to `MX` and `rua/ruf`

	p=none	p=quarantine	p=reject	total
NO MX, DMARC with rua/ruf	204,376	229,777	836,199	1,270,352
NO MX, DMARC without rua/ruf	149,498	35,420	724,429	909,347
MX, DMARC with rua/ruf	3,129,176	1,050,756	1,255,646	5,435,578
MX, DMARC without rua/ruf	5,375,281	1,529,969	1,449,045	8,354,295
Total	8,858,331	2,845,922	4,265,319	15,969,572

DMARC has two main features to protect a domain name against spoofing attacks:

- By configuring DMARC with restrictive handling policies (i.e., `p=quarantine` or `p=reject`), emails failing the DMARC check mechanism may not reach the destination. Table 2 shows that 7,111,241 (44.5%) domain names having DMARC choose this type of protection.

[9] https://dmarc.org/2016/07/common-problems-with-dmarc-records/

- By configuring a DMARC policy with reporting options (i.e., `rua` and/or `ruf`), domain owners can receive alerts regarding any attempt to spoof their domains. 6705930 (42%) domain names having DMARC (see Table 2) opt-in for receiving aggregate and/or failure reports.

When a domain name has no `MX` record (no mail server), it means that no legitimate emails are expected to be sent on behalf of that domain. As a result, it may not be effective for the domain owners to have a DMARC record with `p=none`, which would be less restrictive than `p=reject`.

While domains without `MX` records represent 16.6% of active domains with valid DMARC, 41.7% of them do not use the reporting system, and 149,498 domains choose the handling policy `p=none` (see Table 2). If the domain owners aim to monitor the distribution of malicious emails and have no `MX` records, they can achieve this goal by employing the `rua` or `ruf` mechanisms (204,376 domain names).

For example, the Google domain names `googletagmanager.com` and `goo.gl` do not have `MX`, and contain the SPF record `"v=spf1 -all"` indicating that no server is authorized to send emails on behalf of that domain name. It also contains the following DMARC record:

`"v=DMARC1; p=reject; rua=mailto:mailauth-reports@google.com"`

that specifies the policy of `p=reject` and an aggregate reports to be sent to `mailauth-reports@google.com`.

4.2 Application of DMARC Tags

We next analyze the occurrence frequency of DMARC tags and their content, which provides insight into the behavior of email receivers expected by domain owners with respect to spoofed emails.

Figure 4 shows that domain owners tend to specify tags with default values, even if there is no need to state them explicitly. For instance, most domain owners specify the `pct` tag to 100 (default value), indicating the percentage of emails that should be subject to the DMARC handling policy.

The `fo` tag has a default value of `'0'` that requests the receiver generate a DMARC failure report when SPF and DKIM are not aligned. As shown in Figure 4, when it is present, 75% of the `fo` values differ from `'0'`. It is important to note that the `fo` tag is only useful when a `ruf` tag is present. However, our analysis shows that 33.45% of the DMARC records with an explicit `fo` tag do not have a corresponding `ruf` tag, which may indicate that their domain owners have misunderstood the meaning of `fo` tag and its relationship with `ruf`.

Similarly, the `pct` tag, whose default value is 100, is unnecessary when the domain owner specifies `p=none` since no action (contrary to `reject` or `quarantine`) needs to be taken if DMARC check fails.

However, about 42% of domain owners who use the `pct` tag also set `p=none`. This error may stem from misunderstanding the DMARC mechanism, but it does not interfere with the DMARC check mechanism.

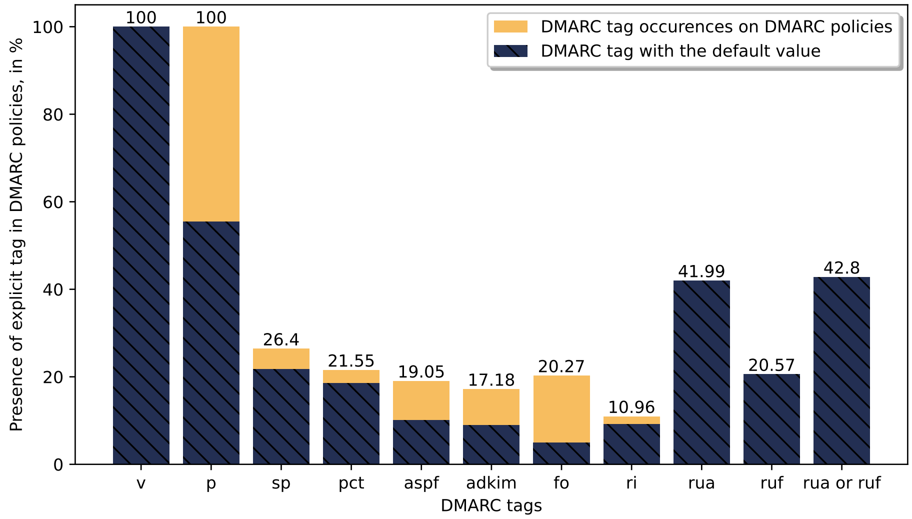

Fig. 4: Occurrence frequency of DMARC tags

In contrast, the `adkim` and `aspf` tags are only present in 16.91% of all DMARC records. However, when considering only the domains with `p=quarantine` or `p=reject`, over 28.63% of domain names specify either `adkim` or `aspf`, which suggests that the administrators of the domains with `p!=none` have more DMARC expertise because the risks of misconfiguration are not negligible.

Lastly, as many as 42% and 21% of domain names have at least one URI in the `rua` and `ruf` tags, respectively, which shows that domain owners with DMARC enabled want to receive DMARC reports. In the following section, we analyze the recipients of aggregate and failure reports as specified in the `ruf` and `rua` tags.

4.3 Statistics on DMARC Report Receivers

We have gathered statistics on DMARC report receivers to identify the main stakeholders involved in report processing. Over 6.8 M domain name owners have expressed interest in receiving at least one type of reports, with a combined total of 11.7 M email addresses, 4.0 M of which are unique. Figure 5 presents the proportion of the registered domain names in the email addresses. We can observe that the first five domain names alone represent 30% of all email addresses present in the DMARC records. We identify three distinct categories of report receivers:

- Third-party email security services such as Proofpoint,[10] Agari,[11] or Mimecast.[12]
- Individuals or organizations who receive DMARC reports to their personal email addresses (e.g., `gmail.com` or `163.com`).
- Hosting providers and domain registrars that provide email systems for their clients such as `dhosting.pl`.

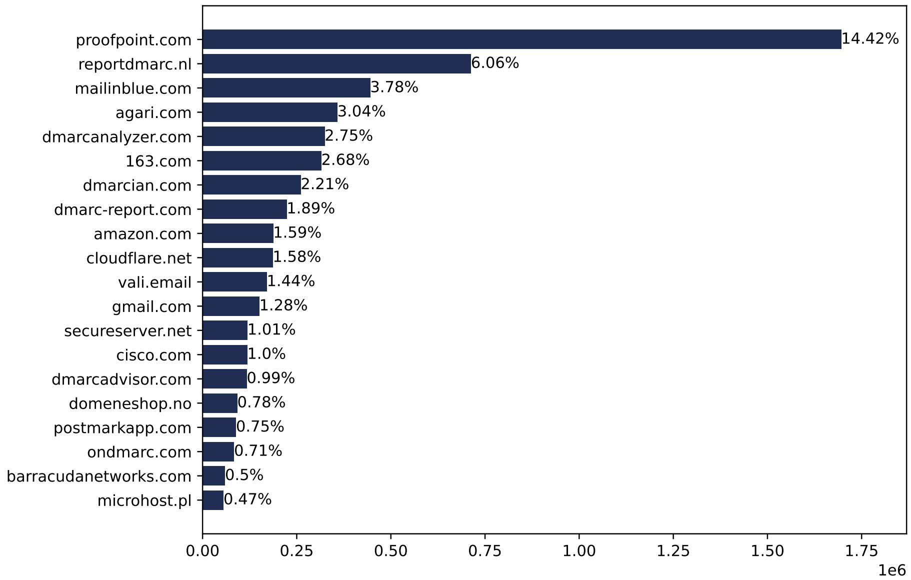

Fig. 5: Registered domain names in the `rua` and `ruf` tags

As Table 1 indicates, M_4 results show that 25% of external verifications failed. Due to this misconfiguration, one million domain names have at least one URI that is not supposed to receive any report as per the RFC specification [28]. Upon aggregating the email addresses specified in the tags `rua` or `ruf` that failed external verification, we have identified approximately 195 K domain names in this category.

We have noticed that some DMARC third-party services choose to accept reports from any domain. For instance, Agari returns `"v=DMARC1;"` for any DNS `TXT` query for any domain name under the `*._report._dmarc.agari.com` wildcard. When querying the `TXT` record for domains such as `example.tld._report._dmarc.agari.com` or `jhdgvr3zt4wcsa._report._dmarc.agari.com` (where `example.tld` is not a valid domain and `jhdgvr3zt4wcsa` is a random string), the returned result is `"v=DMARC1;"`. This result does not seem to be true for Proofpoint, the largest third-party email security services provider (see Figure 5).

10 `https://www.proofpoint.com/`
11 `https://www.agari.com/`
12 `https://dmarcanalyzer.com/`

We have also observed that three invalid email addresses: `address@yourdomain.com`, `me@example.com`, and `youremailaddress@yourdomain.com` appear in more than 29 K records, and each of them is included in the guide for setting up DMARC.[13] [14] [15]

It is interesting to see the distribution of email addresses used for receiving DMARC reports and the dominance of a few third-party email security services, ESPs, and hosting providers/domain registrars. An issue of concern is the significant number of domains incorrectly set up, which should not receive reports due to failing email verification processes.

4.4 Popular Domains

The owners of popular domains have more resources for securing systems and are more susceptible to email spoofing. Therefore, it is reasonable to expect that they deploy DMARC to a larger extent than other domains.

To explore this hypothesis, we have analyzed the DMARC deployment of 1 M most popular domains in the Tranco top site ranking [38]. Figure 6 presents the proportion of the following features characterizing DMARC deployment: a `TXT` record for `_dmarc`, a valid DMARC record, the presence of the `rue/ruf` tag, strict (`p=reject`) handling policy, and external email verification errors for all domains.

As expected, as domain popularity increases, the proportion of valid DMARC records also tends to rise. This trend is accompanied by an increase in the number of domains with the `p=reject` policy, which suggests that more popular domains tend to have a higher confidence level in their DMARC deployment and stricter policies in place.

Nevertheless, the proportion of invalid DMARC records (a `TXT` record present for `_dmarc` but with an invalid DMARC record) remains stable regardless of the popularity rank. Figure 6 indicates that among the most popular domains, there are fewer domain names with active email addresses (no `MX` records). This outcome suggests that large companies may use different domain names for their web presence, which are more popular, and other domain names for email communication.

Although the `ruf` reporting tag is less commonly used than `rua`, it is more prevalent among popular domains. On average, 75% of domain names with DMARC in the top 1 million have at least one reporting tag. In contrast, 42.8% of all domain names have it. This percentage decreases as the domain rank decreases, and the number of email verification errors tends to increase for less popular domain names. The deployment of DMARC, stringent handling policies, the presence of reporting tags, and external destination errors appear to be correlated with the importance of a domain name, which suggests that popular

[13] https://proton.me/support/custom-domain-google

[14] https://help.elasticemail.com/en/articles/2303947-the-dmarc-generator-tool

[15] https://wpmailsmtp.com/how-to-create-dmarc-record/

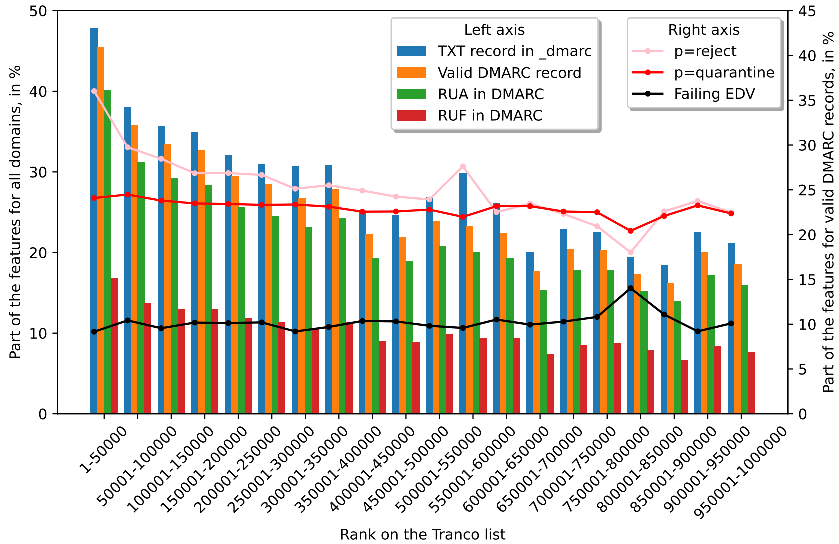

Fig. 6: Features of DMARC deployment for popular domains in Tranco

domains allocate more resources to DMARC, as they have greater incentives to implement DMARC rigorously, given that their domain names are at higher risk of spoofing attacks.

5 Time Evolution of the DMARC Use and Deployment

Table 3 presents the DMARC statistics from related work and the summary of our measurements. The first section of this table, with the measurements based on domain ranking lists (such as Alexa or Tranco), illustrates the trends in DMARC adoption and handling policies since 2014. The results indicate a rising trend in DMARC adoption, with over 25% of popular domains currently having valid DMARC records. The second part of Table 3, which includes a broad sample of the overall domain population, supports similar conclusions with caution due to differences in sample sizes and domain coverage.

As shown in Table 3, our two measurements reveal a decline in `p=none` policies from 67.7% to 55.5%. To understand the changes over a year, we have proceeded with a comparative analysis of our two sets of measurements. Figure 7 illustrates the differences in reporting policies between the two years, allowing us to gain insights into how DMARC adoption and handling policies have evolved.

While the domains included in the measurement M_7 conducted in 2023 (refer to Figure 7 and Table 1) but not present in the measurement M_3 from 2022 constitute 41% of the 2023 datasets, they contribute to 48% of quarantine or reject policies. Therefore, the adoption of more restrictive policies can be attributed to the new DMARC domain names.

Table 3: DMARC related data (related work and our measurements)

Dataset				DMARC Statistics				
Source	Study	Date	Size	Adoption Rate	p tag value none	quarantine	reject	RUA or RUF
Popularity list	Gojmerac et al. [18]	2014-08	677K	0.5%	71.6%	8.0%	20.5%	-
	Tatang et al. [46]	2015-01	1M	1.0%	75.2%	8.2%	16.5%	-
	Durumeric et al. [13]	2015-04	792K	1.1%	72.6%	8.0%	19.4%	-
	Szalachowski et al. [44]	2016-08	100K	7.4%	-	-	-	-
	Hu et al. [21]	2017-10	1M	4.6%	77.6%	10.1%	12.3%	-
	Tatang et al. [46]	2018-12	1M	7.2%	76.1%	11.0%	12.9%	-
	Tatang et al. [46]	2020-05	1M	11.5%	68.5%	15.9%	15.6%	-
	Yajima et al. [51]	2022-02	1M	19.4%	-	-	-	-
	Our parser	2022-09	1M	21.4%	55.3%	21.2%	23.5%	74.6%
	Czybik et al. [8]	2023-05	1M	22.6%	-	-	-	-
	Our parser	2023-10	1M	25.1%	50.9%	23.0%	26.1%	74.9%
Broader source	Portier et al. [39]	2018-01	336M	0.0%	75.2%	7.2%	14.4%	48.9%
	Maroofi et al. [34]	2020-09	236M	0.1%	39.6%	9.3%	41.0%	-
	Nosyk et al. [36]	2022-01	251M	3.3%	49.7%	11.2%	37.1%	-
	DMARC.org [16]	2022-06	-	-	68.2%	12.1%	19.6%	-
	Our parser	2022-09	257M	4.5%	67.7%	14.0%	24.3%	43.4%
	Ashiq et al. [2]	2023-01	89M	6.6%	39.6%	9.3%	41.0%	49.0%
	Our parser	2023-10	280M	5.4%	55.5%	17.8%	26.7%	42.9%

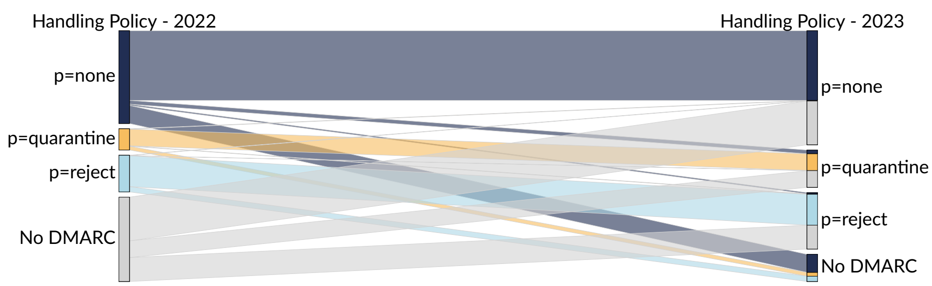

Fig. 7: DMARC evolution between 2022 and 2023

Ovrall, the DMARC population is growing and the trend towards adopting more restrictive handling policies primarily stems from new DMARC domains. However, the influence of aggregate reports on policy modifications remains an open question.

The Parallel Sets Chart in Figure 8 visually illustrates the modifications made to the `p` and `rua` tags within a one-year timeframe. The two colors represent the presence or absence of `rua` tags in DMARC records in 2023. To ensure consistent data in both sets, we undertook specific steps when dealing with domain names in M_7 that were not present in M_3. For these domain names, we collected the registration information and then excluded those that had been registered prior to the previous scans.

In the past, 59% of domain names had `rua` tags, but this proportion decreased to 47% (①). Almost 80% of the modifications occurred when domain names had the `p=none` handling policy in 2022 (②). Among them, 64.1% changed to

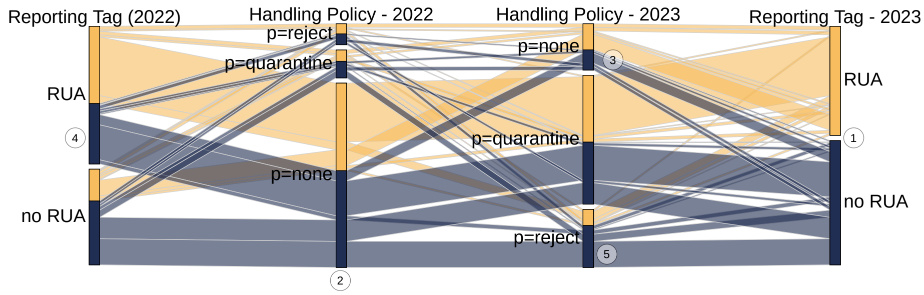

Fig. 8: DMARC evolution between 2022 and 2023. Domains with modified `p` or `rua` tags.

`p=quarantine`, 21.8% moved to `p=reject`, and 14.1% maintained `p=none` while modifying the `rua` tag. When only the `rua` tags is modified (③), 39.2% names added the tag and 60.3% removed them.

Notably, 28.4% of domains with `rua` tags (④) removed the tag while adopting a more restrictive handling policy.

Surprisingly, 54.7% of the domain name that had `p=none` and move to `p=reject`, did not have `rua` tags (⑤). Globally, 35.8% of domain names that have toughened their policies did not have a `rua` tag, which is contradictory to the hypothesis that domain names use the reporting system to transition to more strict policies.

In total, 48.2% of all domain names displayed unexpected behavior: they removed the `rua` tag without adopting more restrictive policies or have adopted more strict policies without having `rua` tags.

When looking at global measurements and related work, it becomes evident that DMARC adoption is on the rise, and restrictive policies such as `p=quarantine` and `p=reject` are becoming more prevalent. However, this growth in handling policies is largely driven by newcomers to the DMARC ecosystem. Older DMARC domains do not appear to be inclined to change their handling policies, suggesting that the reporting system may be ineffective.

6 Vulnerability Notification Campaign

During the analysis of the M_4 external verification process, we have found email addresses specified in `rua` or `ruf` tags belonging to unregistered domain names. So, a malicious actor can register such domain names and configure the support for receiving DMARC reports.

Having unauthorized access to DMARC reports raises several significant threats to an organization. The primary concern is the potential compromise of user communication. If an attacker successfully acquires forensic reports, it may contain the original messages exchanged among users, potentially exposing sensitive information. Additionally, access to aggregate reports allows attackers to gather insights into the organization email infrastructure and communication patterns. This information can be exploited to enhance the effectiveness of impersonation. Lastly, aggregate reports include details about the entities with whom the organization communicates, revealing potential trust relationships. Exploiting this information, attackers can pinpoint weaknesses in anti-spoofing protocols among trusted entities, thereby optimizing their strategies for targeting the organization. This section presents the details on how we have detected vulnerable domain names and how we have contacted their owners. We also present the results of our notification campaign.

The measurement M_4 results in 150 K domain names for which the external verification query returns `NXDOMAIN`, which indicates that the domain name parts of the emails in `rua` or `ruf` do not exist. However, a non-existing subdomain is not enough to decide whether the domain is not registered. Therefore, we have performed an `SOA` query for all organizational domain names. 7,462 of the queries have returned `NXDOMAIN` or empty `SOA`. Then, we have used the WHOIS and RDAP protocols to retrieve the registration information of these domain names. Since certain TLDs, like `.es`, do not offer public WHOIS information, we cannot determine the availability of a given domain name for registration. To avoid sending unsolicited emails, we have selected only the domains for which the WHOIS or RDAP query succeeded. In total, we have found 7,286 domains available for registration, leading to 9,142 vulnerable domain names.

Next, we needed to find a way to contact the owners of the domain names. First, the DMARC record may contain several email addresses in `rua` and `ruf`. Even if the email addresses are used to monitor DMARC reports, they can be an adequate means of contact. Out of 2,458 email addresses in the DMARC record, 799 of them belong to the same organizational domain (which we label as *direct contact*) of the vulnerable domain and 1,659 did not (which we label as *indirect contact*). Second, the WHOIS or RDAP answers may contain the email addresses of the domain registrant or administrator. Even if Ferrante et al. showed that GDPR makes this process less relevant [16], we have obtained the email addresses form WHOIS or RDAP for 1,674 domain names (which we label as *WHOIS contact*). Finally, the email addresses from DMARC and WHOIS or RDAP only represent `3,584` domains. As a consequence, we have also generated email addresses according to the RFC 2142 [6] for the remaining 5,558 domains without WHOIS/RDAP contact, direct, or indirect contact. RFC 2142 specifies the email addresses to be used for contacting common services of an organization such as `common-services@domain`. We have chosen to contact `security@domain` and `admin@domain` (which we label as *RFC contact*).

We have grouped the recipients according to our labels: *direct*, *indirect*, *WHOIS*, and *RFC contacts*, and send emails based on the corresponding templates. As indirect contacts (any email address not directly related to the organizational domain name) may appear in multiple DMARC records, instead of sending an email for each vulnerable domain name, we have sent a list of the vulnerable domains. For multiple WHOIS contacts, we only send a single email. If the emails appear in the same mailbox, they will be shown as one email because

```
dmarc-version   = "v" *WSP "=" *WSP %x44 %x4d %x41 %x52 %x43 %x31
dmarc-request   = "p" *WSP "=" *WSP ("none" / "quarantine" / "reject" )
```

Fig. 9: Extract of the DMARC record ABNF definition from RFC 7489

of the unique email id. However, a domain owner might have been contacted through multiple channels (WHOIS, direct, and indirect)

We have sent 9,218 emails in October 2022. 4,540 emails bounced back, and among them, 4,098 lack any existing recipient. Specifically, 57 WHOIS, 632 direct, 52 indirect, and 3357 RFC. Additionally, 263 messages sent to WHOIS contacts faced delivery challenges and generated automated responses as they were protected by GDPR masking. As a result, we managed to deliver the email to at least one of the contacts for 4,582 domain names. 89 administrators have responded regarding the notifications.

Two weeks after sending the notifications, we re-scanned the vulnerable domain names. The owners of 185 domains removed their DMARC records, 519 modified their DMARC records but 19 were still vulnerable, and 11 added new DMARC records resulting in an invalid DMARC record. Our notification campaign resulted in 685 no longer vulnerable domain names, which makes the remediation rate of 7.5% for the total vulnerable domain names and 15% for the domain names that received an email.

7 Parsing DMARC records

The DMARC record specification and its grammar are outlined in RFC 7489, described using both Augmented Backus-Naur Form (ABNF) and the RFC itself. However, the general public may not be familiar with the ABNF specification or all the finer details presented in the RFC. To address this issue, numerous documentation and services have been created to assist users in creating and verifying their DMARC records. Nevertheless, we have encountered instances for which certain services do not adhere to all the specifications or potentially making them susceptible to Cross-Site Scripting (XSS) vulnerabilities.[17] To investigate this aspect further, we have tested various checkers with different corner cases presented below:

- `space`: in the ABNF rule for `dmarc-version`, the `'='` character is surrounded by `"*WSP"` (white-space or tab), allowing `"v = DMARC1; p=none"` to be a valid DMARC record.
- `case`: in contrast to the `dmarc-version` rule, where every character in the string 'DMARC1' is considered a terminal value, the `dmarc-request` is not case-sensitive. Therefore, `"v=DMARC1; p=ReJeCt"` is recognized as a valid DMARC record.

[17] https://6point6.co.uk/insights/xss-bugs-on-dmarc-checking-sites/

- `case-tag`: similar to the `case` scenario, all tags are not case-sensitive, making `"V=DMARC1; p=none"` a valid DMARC record.
- `xss`: while some DMARC tools assist users in verifying their records, certain tools display the content of the record. If this record is not correctly escaped, it can potentially enable attackers to execute XSS. For example, the record `"v=DMARC1;p=<script>alert('This is an XSS test')</script>;"` will show an alert box in the vulnerable websites.
- `dup`: RFC 6376 specifies that duplicate tags are not allowed in a `tag-list` (see Appendix, Figure 16). However, the handling of this corner case is not clearly defined in RFC 7489. We have brought this issue to the attention of the IETF and it is under consideration.
- `u-tag`: RFC 7489 states that any unknown tags must be ignored. For instance, the record `'v=DMARC1; p=reject; foo=bar;'` is considered a valid DMARC record.
- `p-down`: the 6th bullet in the policy discovery section of RFC 7489 specifies that a record with an invalid `p` or `sp` tag, but with a `rua` containing at least one valid URI, should be interpreted as a record with `p=none` (see Appendix, Figure 13). Therefore, the record `"v=DMARC1;p=reject;sp=error;rua=mailto:rua@example.com"` should be interpreted as `'v=DMARC1;p=none;rua=mailto:rua@example.com'`.

Our experiments involved publishing various DMARC records, as defined earlier, and manually using the DMARC checkers provided by 16 different companies. In the first round of measurements in S1-2022, we identified non-conforming organizations and reached out to them when possible. Four organizations responded and made the necessary changes. As shown in Table 4, four of these organizations were found to be susceptible to XSS vulnerabilities.

During this period, Agari, Dmarcian, and SimpleDmarc did not reply to us and have changed their checker. In October 2023, we re-ran the measurements with the newly discovered corner case `dup`, `u-tag`, and `p-down`. SimpleDmarc has changed its implementation and is vulnerable to XSS. We have contacted the founder of SimpleDmarc and they have fixed the vulnerability. Dmarc360 does not provide a freely accessible DMARC checker anymore.

While it is true that none of the organizations fully adhere to all the RFC 7489 specifications, the `space`, `case`, and `case-tag` rules rely on ABNF knowledge that may not be commonly known by DMARC users. These rules serve to provide relaxed standards to accommodate a wider range of DMARC records. On the other hand, the `dup` and `u-tag` tags hold more significant importance. The `dup` corner case, when not respected, corresponds to a situation in which the email receiver must choose between two handling policies, leading to an undefined behavior. We have contacted the DMARC working group regarding our concerns. The group has indicated that the record should be disregarded. The `u-tag` tag is particularly vital. If the tag list is updated, and the checker is not, the introduction of new tags may cause the checker to reject the records. However, it is noteworthy that none of the checkers adhere to the `downgrade` corner case. It is essential to consider that the `downgrade` feature may be seen as

Table 4: Compliancy of DMARC parsers. ✓: compliant, ✗: not compliant, (✗) fixed the issue after our contact, ?: Behavior cannot be defined.

	2022				2023						
	space	case	case-tag	xss	space	case	case-tag	xss	dup	u-tag	p-down
Online DMARC checker											
Agari	✓	✓	✗	✗	✓	✓	✗	✓	✗	✓	✗
Dmarc360	✗	✓	✓	✓	?	?	?	?	?	?	?
Dmarcadvisor	✓	✗	(✗)	✓	✓	✗	✓	✓	✗	✓	✗
Dmarcanalyzer	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗
Dmarcian	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	?
Dmarcly	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Dnschecker	✗	✗	✓	✓	✗	✗	✓	✓	✗	✓	✗
EasyDmarc	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗
Google	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗
Kdmarc	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
Merox	✓	✓	(✗)	✓	✓	✓	✓	✓	✓	✗	✗
Mxtoolbox	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗
PowerDmarc	(✗)	✓	(✗)	(✗)	✓	✓	✓	✓	✓	✗	✗
Proofpoint	✓	✗	(✗)	(✗)	✓	✗	✓	✓	✓	✓	✗
SimpleDmarc	✗	✓	✗	✓	?	?	?	(✗)	?	?	?
Valimail	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	?
Libraries											
Checkdmarc	(✗)	(✗)	(✗)	-	✓	✓	✓	-	✗	✗	✗
OpenDmarc	-	-	-	-	✓	✓	✓	-	✗	✓	✗
Rspamd	-	-	-	-	✓	✓	✗	-	✗	✓	✗
Our parser	-	-	-	-	✓	✓	✓	-	✓	✓	✓

somewhat far-fetched, and email receivers might not necessarily need to account for this corner case. We have brought up our concerns about these issues to the DMARC working group (details not provided for anonymity reasons).

Table 4 also presents the measurements we conducted on three popular DMARC libraries: OpenDmarc, Rspamd, and checkdmarc. Notably, in related work, only Czybik et al. [8] have disclosed the software they used to parse DMARC records (checkdmarc). In their survey, Ashiq et al. [2] found that one-third of DMARC operators use OpenDMARC. However, none of these libraries met our requirements, particularly for parsing. Consequently, we have developed our own Python ABNF-based DMARC parser, accessible at:

`https://github.com/drakkar-lig/abnf-dmarc-parser`

RFC 7489 provides four different resources for parsing DMARC records (see Appendix, Figures 10, 11, 12, and 13). These resources outline specific rules for parsing DMARC records. The first statement regarding parsing specifies that DMARC records follow the "tag-value" syntax defined in DKIM, and any unknown tags must be ignored. The second statement highlights that a DMARC policy record must adhere to the ABNF, with the 'v' and 'p' tags appearing first and second, respectively. Unknown tags must be ignored, and certain syntax errors should be discarded. The third statement provides the ABNF rules for DMARC records, while the last statement addresses the 'p-down' corner case. To ensure proper parsing, we initially apply the rules outlined in RFC 6376 (see Appendix, Figures 14, 15, and 16) to ensure that the strings match the 'tag-list' ABNF, without duplicate. Subsequently, we apply the RFC 7489 ABNF, ignoring any unknown tags, Finally, we verify that the 'v' and 'p' tags are in the correct order, 'sp' inherit from 'p' if not provided, and verify the `pct` value. This three-step process helps ensure the compliance with the specified parsing rules. As a default behavior, the parser does not apply the 'p-downgrade' corner case, as it is considered optional ('should') rather than mandatory ('must'). This approach aligns with the flexibility provided in the DMARC specifications and ensures that the parser does not enforce this corner case by default.

Ashiq et al. [2] provided their measurement data. We have run our parser with the `follow_dowgrade` option on their data. We have observed slight differences of 1.5% less valid DMARC records according to the value provided in their paper. Unfortunately, we cannot make a direct comparison as they did not provide the parsed data nor their code.

8 Ethical Considerations

To obtain reliable results with minimal interference on the tested systems, we followed the best practices recommended by the measurement community [14,37,12]. We used Google and Cloudflare public resolvers for active measurements and respected the default DNS rate limits. We also randomized our input lists across the IP space and TLDs to avoid sending bulk DNS requests to any single entity, even though most responses are expected to come from Google and Cloudflare DNS caches. Finally, we distributed our scanning activities over several days.

We have enforced contacting each organization having a DMARC checker which was vulnerable to XSS vulnerability before the publication of the article. The only organization that is still vulnerable at the publication time has acknowledged the vulnerability in May 2022. They replied to us that they would 'take a look shortly'. Furthermore, the XSS is performed when a user specifically queries a domain name, not any URL can result directly in an XSS.

Finally, we alerted the domain owners or associated intermediaries about the unregistered domain names we found in the `rua` and `ruf` URIs, to prevent malicious actors from registering them and receiving DMARC reports. Instead of sending multiple emails to the same recipients for each vulnerable domain name, we sent single emails informing each responsible party of all the vulnerable domains.

9 Conclusion

Our measurements reveal potential shortcomings in the understanding and interpretation of the DMARC protocol, as outlined in RFC 7489. None of the organizations we evaluated managed to successfully pass all of our test scenarios. Our analysis of various corner cases reveals that, despite DMARC being a fundamental service, some organizations and open-source projects have either implemented DMARC record parsing tools incorrectly or taken initiatives that deviate from the standards.

Our analysis, which covered 280 million domain names, reveals the following findings regarding the DMARC adoption:

- Out of the 16.4 M domain names containing at least the case-insensitive string 'dmarc' in their TXT record, 150 K were found to be syntactically invalid. Additional 68 K had multiple syntactically valid records, rendering their DMARC invalid.
- Approximately 15.9 million domain names were identified as having valid DMARC records.
- Within this group, one million domain names failed the Email Destination Verification.
- 5.5 million domain names had DMARC records but lacked protections, including reporting options and restrictive handling policies.
- Notably, 35% of domain names that specified an `fo` tag did not have a `ruf` tag (equivalent to 1.1 M domain names).
- Furthermore, 268 K domain names had a `pct` value different from `'100'` while the `p` tag was set to `none`.

Within the realm of popular domains, our observations suggest that administrators of top-ranked domain names demonstrate a better understanding, implementation, and stricter handling policies, potentially linked to the resources dedicated to DMARC. Notably, we have observed that while 42% of domain names with valid DMARC record express a preference for receiving reports, more than 30% opt for the five biggest third-party services to handle these reports, highlighting the complexity of self-management.

To offer a comprehensive overview, our temporal analysis has unveiled that the use of aggregate reports does not display a clear correlation with the changes in handling policies, which indicates that the adjustments in handling policies might not always be directly influenced by the analysis of aggregate reports.

The complexity of standards hinders DMARC deployment and its correct configuration. Improving specifications is an on-going work, for instance, RFC 7489 was published as an informational document and the IETF DMARC working group currently works on an Internet Standards Track for DMARC.[18] The latest accessed version (28) includes modifications to the current DMARC protocol such as the addition or updating of terms and definitions, the introduction of a new process of policy discovery, the removal of the `pct`, `rf`, and `ri` tags, the addition of three new tags, and new RFCs for aggregate and forensic reports.

Nevertheless, it is unlikely that the DMARC version will change, and email receivers will need to ensure backward compatibility between two RFCs. We suggest that a more effective approach would be to define a new DMARC version

[18] `https://datatracker.ietf.org/doc/draft-ietf-dmarc-dmarcbis/`

(`v=DMARC2`) in the new RFC. Specifying DMARC v2 could be an opportunity to deeply redesign the protocol to simplify and clarify its operation. A common belief on DMARC is that the check mechanism would fail if DKIM **or** SPF: Yajima et al. [51] and Ashiq et al. [2] have embraced this misconception. Additionally, the `fo` tag is often misunderstood: as it is often thought to allow the domain owner to indicate the logical operators for the DMARC check mechanism and not the generation of failure reports. Our measurements show that 35% of the DMARC records with `fo` do not have the `ruf` tag, which reveals a misunderstanding of this feature. We think that it is necessary to have more verbal tags and be able to choose the logical operator of the DMARC check mechanism.

Acknowledgments

We thank the reviewers for their valuable and constructive feedback. This work has been partially supported by the French Ministry of Research projects PERSYVAL Lab under contract ANR-11-LABX-0025-01 and DiNS under contract ANR-19-CE25-0009-01.

References

1. Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J., Thomas, M.: Domainkeys identified mail (dkim) signatures. RFC 4871, RFC Editor (May 2007)
2. Ashiq, M.I., Li, W., Fiebig, T., Chung, T.: You've Got Report: Measurement and Security Implications of DMARC Reporting. In: USENIX Security Symposium (2023)
3. Bennett, N., Sowards, R., Deccio, C.: Spfail: Discovering, measuring, and remediating vulnerabilities in email sender validation. In: ACM Internet Measurement Conference. p. 633–646. ACM (2022)
4. Bradner, S.: Key words for use in rfcs to indicate requirement levels. BCP 14, RFC Editor (March 1997)
5. Chen, J., Paxson, V., Jiang, J.: Composition kills: A case study of email sender authentication. In: USENIX Security Symposium. pp. 2183–2199 (2020)
6. Crocker, D.: Mailbox names for common services, roles and functions. RFC 2142, RFC Editor (May 1997)
7. Crocker, D., Hansen, T., Kucherawy, M.: Domainkeys identified mail (dkim) signatures. STD 76, RFC Editor (September 2011)
8. Czybik, S., Horlboge, M., Rieck, K.: Lazy gatekeepers: A large-scale study on spf configuration in the wild. In: ACM Internet Measurement Conference. p. 344–355. ACM (2023)
9. Dan, K., Kitagawa, N., Sakuraba, S., Yamai, N.: Spam domain detection method using active dns data and e-mail reception log. In: Computer Software and Applications Conference (COMPSAC). vol. 1, pp. 896–899 (2019)
10. Deccio, C., Yadav, T., Bennett, N., Hilton, A., Howe, M., Norton, T., Rohde, J., Tan, E., Taylor, B.: Measuring email sender validation in the wild. In: ACM International Conference on Emerging Networking EXperiments and Technologies (CoNEXT). p. 230–242. ACM (2021)

11. Delany, M.: Domain-based email authentication using public keys advertised in the dns (domainkeys). RFC 4870, RFC Editor (May 2007)
12. Dittrich, D., Kenneally, E.: The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research (2012)
13. Durumeric, Z., Adrian, D., Mirian, A., Kasten, J., Bursztein, E., Lidzborski, N., Thomas, K., Eranti, V., Bailey, M., Halderman, J.A.: Neither snow nor rain nor mitm...: An empirical analysis of email delivery security. In: ACM Internet Measurement Conference (IMC). p. 27–39. ACM (2015)
14. Durumeric, Z., Wustrow, E., Halderman, J.A.: ZMap: Fast Internet-Wide Scanning and Its Security Applications. In: USENIX Security Symposium (2013)
15. Fernandez, S., Korczyński, M., Duda, A.: Early detection of spam domains with passive dns and spf. In: Passive and Active Measurement (PAM). pp. 30–49. Springer International Publishing (2022)
16. Ferrante, A.J.: The impact of gdpr on whois: Implications for businesses facing cybercrime. Cyber Security: A Peer-Reviewed Journal **2**(2), 143–148 (2018)
17. Fontana, H.: Authentication failure reporting using the abuse reporting format. RFC 6591, RFC Editor (April 2012)
18. Gojmerac, I., Zwickl, P., Kovacs, G., Steindl, C.: Large-scale active measurements of DNS entries related to e-mail system security. In: IEEE International Conference on Communications (ICC). pp. 7426–7432 (Jun 2015), iSSN: 1938-1883
19. Herzberg, A.: Dns-based email sender authentication mechanisms: A critical review. Computers & Security **28**(8), 731–742 (2009)
20. Hu, H., Peng, P., Wang, G.: Towards understanding the adoption of anti-spoofing protocols in email systems. In: IEEE Cybersecurity Development (SecDev). pp. 94–101 (2018)
21. Hu, H., Wang, G.: Revisiting email spoofing attacks. arXiv preprint arXiv:1801.00853 (2018)
22. Izhikevich, L., Akiwate, G., Berger, B., Drakontaidis, S., Ascheman, A., Pearce, P., Adrian, D., Durumeric, Z.: Zdns: A fast dns toolkit for internet measurement. In: ACM Internet Measurement Conference (IMC). p. 33–43. ACM (2022)
23. Kitterman, S.: Sender policy framework (spf) authentication failure reporting using the abuse reporting format. RFC 6652, RFC Editor (June 2012)
24. Kitterman, S.: Sender policy framework (spf) for authorizing use of domains in email, version 1. RFC 7208, RFC Editor (April 2014)
25. Klensin, J.: Simple mail transfer protocol. RFC 5321, RFC Editor (October 2008)
26. Konno, K., Dan, K., Kitagawa, N.: A spoofed e-mail countermeasure method by scoring the reliability of dkim signature using communication data. In: IEEE Computer Software and Applications Conference (COMPSAC). vol. 2, pp. 43–48 (2017)
27. Kucherawy, M.: Extensions to domainkeys identified mail (dkim) for failure reporting. RFC 6651, RFC Editor (June 2012)
28. Kucherawy, M., Zwicky, E.: Domain-based message authentication, reporting, and conformance (dmarc). RFC 7489, RFC Editor (March 2015)
29. Lever, C., Walls, R., Nadji, Y., Dagon, D., McDaniel, P., Antonakakis, M.: Domain-z: 28 registrations later measuring the exploitation of residual trust in domains. In: IEEE Symposium on Security and Privacy (SP). pp. 691–706 (2016)
30. Liu, D., Hao, S., Wang, H.: All your dns records point to us: Understanding the security threats of dangling dns records. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). p. 1414–1425. ACM (2016)
31. Liu, G., Jin, L., Hao, S., Zhang, Y., Liu, D., Stavrou, A., Wang, H.: Dial "N" for NXDomain: The Scale, Origin, and Security Implications of DNS Queries to

Non-Existent Domains. In: ACM Internet Measurement Conference (IMC). ACM (2023)
32. Malatras, A., Coisel, I., Sanchez, I.: Technical recommendations for improving security of email communications. In: International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). pp. 1381–1386 (2016)
33. Maroofi, S., Korczyński, M., Duda, A.: From defensive registration to subdomain protection: Evaluation of email anti-spoofing schemes for high-profile domains. In: Traffic Measurement and Analysis Conference (TMA) (2020)
34. Maroofi, S., Korczyński, M., Hölzel, A., Duda, A.: Adoption of email anti-spoofing schemes: A large scale analysis. IEEE Transactions on Network and Service Management **18**(3), 3184–3196 (2021)
35. Mori, T., Sato, K., Takahashi, Y., Ishibashi, K.: How is e-mail sender authentication used and misused? In: Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS). pp. 31–37. ACM (2011)
36. Nosyk, Y., Hureau, O., Fernandez, S., Duda, A., Korczyński, M.: Unveiling the weak links: Exploring dns infrastructure vulnerabilities and fortifying defenses. In: IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 546–557. IEEE Computer Society (2023)
37. Partridge, C., Allman, M.: Ethical Considerations in Network Measurement Papers. Commun. ACM **59**(10), 58–64 (sep 2016)
38. Pochat, V.L., Van Goethem, T., Tajalizadehkhoob, S., Korczyński, M., Joosen, W.: Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In: NDSS Symposium (2019)
39. Portier, A., Carter, H., Lever, C.: Security in plain txt. In: Perdisci, R., Maurice, C., Giacinto, G., Almgren, M. (eds.) Detection of Intrusions and Malware, and Vulnerability Assessment. pp. 374–395. Springer International Publishing (2019)
40. Postel, J.B.: Simple mail transfer protocol. STD 10, RFC Editor (August 1982)
41. Scheffler, S., Smith, S., Gilad, Y., Goldberg, S.: The unintended consequences of email spam prevention. In: Passive and Active Measurement (PAM). pp. 158–169. Springer International Publishing (2018)
42. Schlamp, J., Gustafsson, J., Wählisch, M., Schmidt, T.C., Carle, G.: The abandoned side of the internet: Hijacking internet resources when domain names expire. In: Traffic Measurement and Analysis Conference (TMA) (2015)
43. Shukla, S., Misra, M., Varshney, G.: Forensic analysis and detection of spoofing based email attack using memory forensics and machine learning. In: Security and Privacy in Communication Networks. pp. 491–509. Springer Nature Switzerland (2023)
44. Szalachowski, P., Perrig, A.: Short paper: On deployment of dns-based security enhancements. In: Financial Cryptography and Data Security. pp. 424–433. Springer International Publishing (2017)
45. Tatang, D., Flume, R., Holz, T.: Extended abstract: A first large-scale analysis on usage of mta-sts. In: Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA). pp. 361–370. Springer International Publishing (2021)
46. Tatang, D., Zettl, F., Holz, T.: The evolution of dns-based email authentication: Measuring adoption and finding flaws. In: International Symposium on Research in Attacks, Intrusions and Defenses (RAID). p. 354–369. Association for Computing Machinery (2021)
47. Vissers, T., Barron, T., Van Goethem, T., Joosen, W., Nikiforakis, N.: The wolf of name street: Hijacking domains through their nameservers. In: ACM SIGSAC

Conference on Computer and Communications Security (CCS). p. 957–970. ACM (2017)
48. Wang, C., Wang, G.: Revisiting email forwarding security under the authenticated received chain protocol. In: ACM Web Conference (WWW). p. 681–689. ACM (2022)
49. Wang, C., Shen, K., Guo, M., Zhao, Y., Zhang, M., Chen, J., Liu, B., Zheng, X., Duan, H., Lin, Y., et al.: A large-scale and longitudinal measurement study of {dkim} deployment. In: USENIX Security Symposium. pp. 1185–1201 (2022)
50. Wong, M., Schlitt, W.: Sender policy framework (spf) for authorizing use of domains in e-mail, version 1. RFC 4408, RFC Editor (April 2006)
51. Yajima, M., Chiba, D., Yoneya, Y., Mori, T.: A first look at brand indicators for message identification (bimi). In: International Conference on Passive and Active Network Measurement. pp. 479–495. Springer (2023)

10 Appendix

```
DMARC records follow the extensible "tag-value" syntax for
DNS-based key records defined in DKIM [DKIM].
Section 11 creates a registry for known DMARC tags and registers
the initial set defined in this document.  Only tags defined in
this document or in later extensions, and thus added to that
registry, are to be processed; unknown tags MUST be ignored.
```

Fig. 10: RFC 7489 Extract - 6.3. General Record Format

```
A DMARC policy record MUST comply with the formal specification
found in Section 6.4 in that the "v" and "p" tags MUST be present
and MUST  appear in that order.  Unknown tags MUST be ignored.
Syntax errors in the remainder of the record SHOULD be discarded
in favor of default values (if any) or ignored outright.
```

Fig. 11: RFC 7489 Extract - 6.3. General Record Format

```
dmarc-record    = dmarc-version dmarc-sep  [dmarc-request]
[dmarc-sep dmarc-srequest] [dmarc-sep dmarc-auri]
[dmarc-sep dmarc-furi] [dmarc-sep dmarc-adkim]
[dmarc-sep dmarc-aspf] [dmarc-sep dmarc-ainterval]
[dmarc-sep dmarc-fo] [dmarc-sep dmarc-rfmt]
[dmarc-sep dmarc-percent] [dmarc-sep]
; components other than dmarc-version and
; dmarc-request may appear in any order
```

Fig. 12: RFC 7489 Extract - 6.3. General Record Format

```
6.  If a retrieved policy record does not contain a valid "p" tag
   , or contains an "sp" tag that is not valid, then:
  1.  if a "rua" tag is present and contains at least one
     syntactically valid reporting URI, the Mail Receiver SHOULD
     act as if a record containing a valid "v" tag and "p=none"
     was retrieved, and continue processing;
  2.  otherwise, the Mail Receiver applies no DMARC processing to
     this message.
```

Fig. 13: RFC 7489 Extract - 6.6.3. Policy Discovery

```
DKIM uses a simple "tag=value" syntax in several contexts,
including in messages and domain signature records.
Values are a series of strings containing either plain text,
"base64" text (as defined in [RFC2045], Section 6.8),
"qp-section" (ibid, Section 6.7), or "dkim-quoted-printable"
(as defined in Section 2.11). The name of the tag will
determine the encoding of each value.  Unencoded semicolon
(";") characters MUST NOT occur in the tag value, since that
separates tag-specs.
```

Fig. 14: RFC 6376 Extract - 3.2. Tag=Value Lists

```
tag-list  =  tag-spec *( ";" tag-spec ) [ ";" ]
tag-spec  =  [FWS] tag-name [FWS] "=" [FWS] tag-value [FWS]
tag-name  =  ALPHA *ALNUMPUNC
tag-value =  [ tval *( 1*(WSP / FWS) tval ) ]
                 ; Prohibits WSP and FWS at beginning and end
tval      =  1*VALCHAR
VALCHAR   =  %x21-3A / %x3C-7E
                 ; EXCLAMATION to TILDE except SEMICOLON
ALNUMPUNC =  ALPHA / DIGIT / "_"
```

Fig. 15: RFC 6376 Extract - 3.2. Tag=Value Lists

```
Tags MUST be interpreted in a case-sensitive manner.  Values MUST
be processed as case sensitive unless the specific tag
description of semantics specifies case insensitivity.
Tags with duplicate names MUST NOT occur within a single tag-list
; if a tag name does occur more than once, the entire tag-list is
invalid.
Whitespace within a value MUST be retained unless explicitly
excluded by the specific tag description.
Tag=value pairs that represent the default value MAY be included
to aid legibility.
Unrecognized tags MUST be ignored.
```

Fig. 16: RFC 6376 Extract - 3.2. Tag=Value Lists