

BEST PRACTICES FOR EMAIL AUTHENTICATION

We recommend you always set up these email authentication methods for your domain:

- SPF helps servers verify that messages appearing to come from a particular domain are sent from servers authorized by the domain owner.
- DKIM adds a digital signature to every message. This lets receiving servers verify that messages aren't forged, and weren't changed during transit.
- DMARC enforces SPF and DKIM authentication, and lets admins get reports about message authentication and delivery.

An abstract graphic at the top of the slide featuring a series of overlapping, wavy bands in shades of orange, red, yellow, and green, set against a black background.

ENSURE MAIL DELIVERY & PREVENT SPOOFING (SPF)

PROTECT AGAINST FORGED EMAILS & MAKE SURE MESSAGES AREN'T MARKED AS SPAM.

SPF RECORD

Sender Policy Framework (SPF) is an email authentication method that specifies the mail servers authorized to send email for your domain. SPF helps protect your domain from spoofing, and helps ensure that your messages are delivered correctly. Mail servers that get mail from your domain use SPF to verify that messages that appear to come from your domain actually are from your domain.

WHAT EFFECT DOES AN SPF RECORD HAVE?

- SPF help prevents spoofing—Spammers can forge your domain or organization to send fake messages that appear to come from your organization. This is called *spoofing*. Spoofed messages can be used for malicious purposes, for example to communicate false information, to send out harmful software, or to trick people into giving out sensitive information. SPF helps receiving servers verify that mail sent from your domain is actually from your organization, and is sent by a mail server authorized by you.
- SPF helps deliver messages to recipients' inboxes—SPF helps prevent messages from your domain from being delivered to spam. If your domain doesn't use SPF, receiving mail servers can't verify that messages appearing to be from your domain actually are from you. Receiving servers might send valid messages to recipients' spam folders, or might reject valid messages.

SPF MECHANISMS

The following mechanisms define which IP addresses are allowed to send mail from the domain:

- a
- mx
- ip4
- ip6
- exists

A mail server will compare the IP address of the sender against the IP addresses defined in the mechanisms. If the IP address matches one of the mechanisms in the SPF record then follow the result handling rule. The default handling rule is + or pass.

The **include** mechanism allows you to authorize hosts outside of your administration by specifying their SPF records.

The **all** mechanism matches any address. This is usually used as the last mechanism which defines how to handle any sender IP that did not match the previous mechanisms.

All mechanisms may specify qualifiers for how to handle a match:

- + for pass
- for fail
- ~ for soft fail
- ? for neutral

As previously mentioned, the default handling rule is pass, which is the same as the + qualifier.

HOW DO I CREATE AN SPF RECORD?

- Start by gathering a list of all your domains, as each SPF record refers to a specific domain. Be sure to include inactive (or “parked”) domains that don’t send email in order to protect them from abuse as well.
- You will also need to identify everything that sends email from your domain(s), including *sources* (third-parties) that send emails on behalf of your domain. This includes:
 - Mail Servers (both web-based like Gmail or via your ISP and in-office like Microsoft Exchange)
 - ESPs (Email Service Providers – companies that provide email marketing/bulk email services)
 - Miscellaneous services (e.g., support/ticketing systems, payment providers, e-merchant services, etc.)

v=spf1 ip4:40.113.200.201 ip6:2001:db8:85a3:8d3:1319:8a2e:370:7348 include:thirdpartydomain.com ~all

1. Start with the SPF version **v=spf1**. This indicates that it is an SPF record. It will always be v=spf1, as other SPF versions have been discontinued.
2. The SPF version tag should be followed with all IP addresses that are authorized to send email on behalf of your domain.
For example: v=spf1 *ip4:40.113.200.201 ip6:2001:db8:85a3:8d3:1319:8a2e:370:7348*

3. Next comes the “include” statement, which is needed for every third-party organization that sends email on your behalf.

For example:

```
v=spf1  
ip4:40.113.200.201 ip6:2001:db8:85a3:8d3:1319:8a2e:370:7348 include:thirdpartydomain.com
```

You should consult with these third parties to discover which domain to use as a value here. Also, ESPs typically publish SPF records for sending domains on your behalf, so you will want to verify with them as well.

4. The end of the SPF record is the “all” tag. It is important because it indicates what policy and how strictly it should be applied when a receiving server detects a server which is not listed (authorized) in your SPF record.
The “all” tag has the following basic options:
 - all – (fail) non-authorized emails will be rejected*.
 - ~all – (softfail) non-authorized emails will be accepted but marked*.
 - +all – this tag allows any server to send email from your domain, so we advise strongly against it.

For example:

```
v=spf1  
ip4:40.113.200.201 ip6:2001:db8:85a3:8d3:1319:8a2e:370:7348 include:thirdpartydomain.com all
```

- After defining your SPF record your record might look something like this:

```
v=spf1 ip4:34.243.61.237 ip6:2a05:d018:e3:8c00:bb71:dea8:8b83:851e  
include:thirdpartydomain.com -all
```

- For domains that aren't sending email, we recommend you to publish the following record `v=spf1 -all`

Please keep in mind that your SPF record cannot be over 255 characters and has a maximum of 10 include tags, also known as “lookups”. Please note that the ‘nested lookups’ will also count. If a record has an A and MX lookup, these will both count as lookups for your domain.

SPF RECORD LIMITATIONS

Each fully-qualified name may have at maximum one SPF record, defined as a TXT record or as an SPF record type.

There are various limitations on the number of items and lookups permitted in an SPF record:

- SPF records may not have more than 10 mechanisms that require DNS lookups. These are the **include**, **a**, **mx**, **ptr**, and **exists** mechanisms.
- When evaluating the **mx** mechanism, the number of MX records queried is included in the overall limit of DNS lookups.
- Each **mx** mechanism must not result in querying more than 10 address records.
- The **ptr** mechanism is also included in the overall limit. Each **ptr** must not result in querying more than 10 address records.


PUBLISH YOUR SPF RECORD INTO YOUR DNS

Finally, after defining your SPF record it's time to publish the record into your DNS. Doing so, mail receivers like (Gmail, Hotmail and others) can request it. An SPF record needs to be published into your DNS by your DNS manager. This can be an internal role in your organization, you can have access to a dashboard provided by your DNS provider yourself or you can ask your DNS provider to publish the record.

CHECKING SPF RECORD

mxtoolbox.com/SuperTool.aspx

☆ ⚙️ 🌸 Update



PricingToolsDelivery CenterMonitoringProductsSupportLogin

SuperToolMX LookupBlacklistsDMARCDiagnosticsEmail HealthDNS LookupAnalyze HeadersAll Tools

SuperTool Beta7

SPF Record Lookup

spf:link3.net

Find ProblemsSolve Email Delivery Problems

spf


v=spf1 mx ip4:123.200.0.101 ip4:123.200.0.4 ip4:123.200.0.0/29 ip4:203.76.96.6 ip4:203.76.96.118 ip4:203.76.105.178 include:spf.protection.outlook.com include:_


Prefix	Type	Value	PrefixDesc	Description
	v	spf1		The SPF record version
+	mx		Pass	Match if IP is one of the MX hosts for given domain name.
+	ip4	123.200.0.101	Pass	Match if IP is in the given range.
+	ip4	123.200.0.4	Pass	Match if IP is in the given range.
+	ip4	123.200.0.0/29	Pass	Match if IP is in the given range.
+	ip4	203.76.96.6	Pass	Match if IP is in the given range.
+	ip4	203.76.96.118	Pass	Match if IP is in the given range.
+	ip4	203.76.105.178	Pass	Match if IP is in the given range.
+	include	spf.protection.outlook.com	Pass	The specified domain is searched for an 'allow'


[Feedback](#) [Contact](#) [Terms & Conditions](#) [Site Map](#) [API](#) [Privacy](#)


Your IP is: 180.148.212.62
Phone: (866)-MXTOOLBOX / (866)-698-6652 | feedback@mxtoolbox.com
© Copyright 2004-2021, MXToolBox, Inc. All rights reserved.

Free MxToolBox Account
Get 1 Free Monitor*, Email Notifications and Troubleshooting Info

 **Delivery Center**
Real-time insight into the Email Deliverability of you or your 3rd party senders

 **Blacklist Monitoring**
100+ Blacklist Monitored + Delisting Support

 **MailFlow Monitoring**
Round-trip email server monitoring for latency and email deliverability issues

 **Bulk Lookup**
Run Bulk lists of IPs and Domains Blacklist, MX/NS/A Record, GeoIP, & more data

📧 📱 📺

DNS Records for Mail Transaction .pptx - PowerPoint

DKIM (DomainKeys Identified Mail)



WHAT IS A DKIM RECORD?

DKIM (DomainKeys Identified Mail) is an email security standard designed to make sure messages aren't altered in transit between the sending and recipient servers.

- DKIM allows the receiver to check that an email was indeed sent and authorized by the owner of that domain. This is done by giving the email a digital signature. This DKIM signature is a header that is added to the message and is secured with encryption.
- Once receiver (or receiving system) determines that an email is signed with a valid DKIM signature, it's certain that parts of the email among which the message body and attachments haven't been modified. Usually, DKIM signatures are not visible to end-users, the validation is done on a server level.
- Implementing the DKIM standard will improve email deliverability. If you use DKIM record together with DMARC (and even SPF) you can also protect your domain against malicious emails sent on behalf of your domains. DMARC and DMARC Analyzer use both SPF and DKIM. Together they provide synergy and the best result for email security and deliverability.

HISTORY OF DOMAIN KEYS IDENTIFIED MAIL

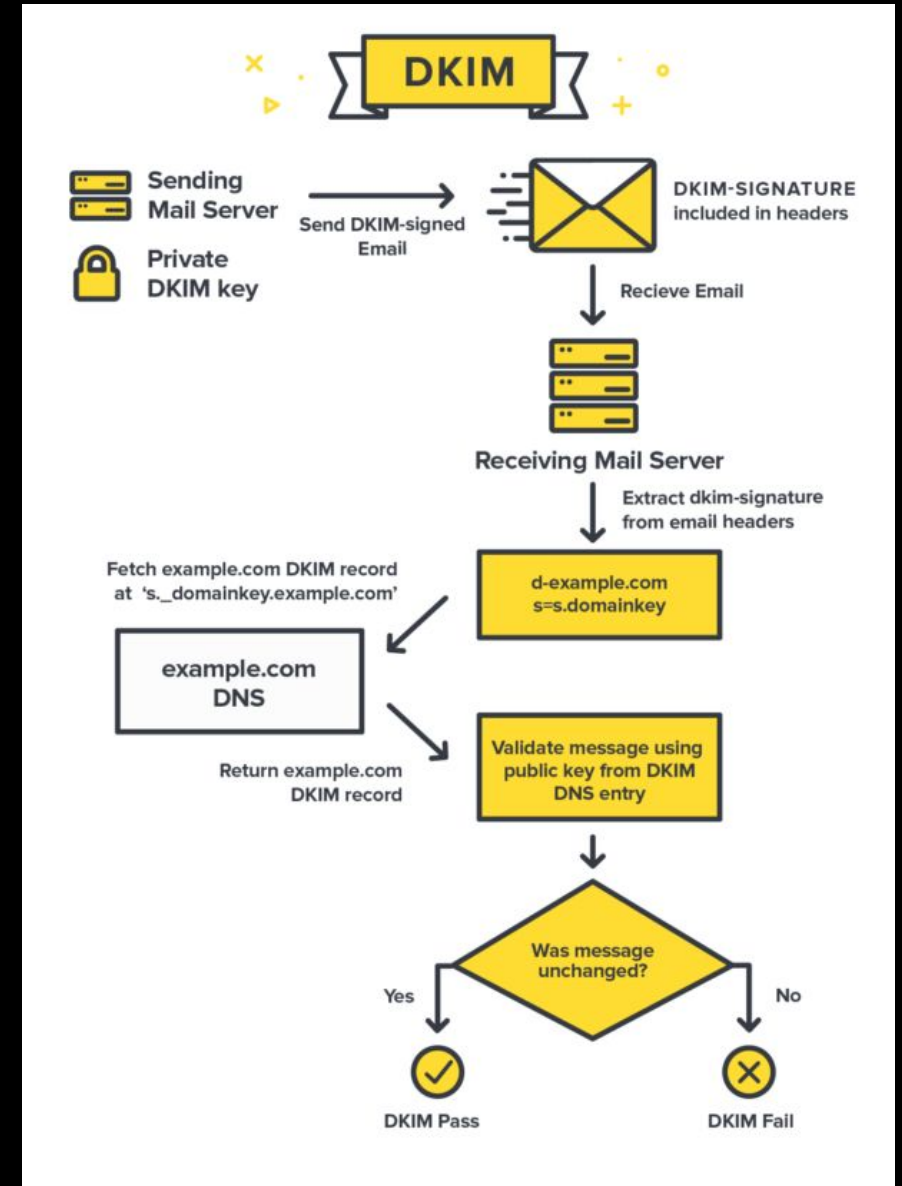
- DKIM was formed by merging two existing specifications Domain Keys (created by Yahoo) and Identified Internet Mail (from Cisco) in 2004.
- It developed into a new widely adopted authentication technique which was also registered as an RFC by the IETF. All leading ISP's (like Google, Microsoft and Yahoo) check incoming mail for DKIM signatures.

DKIM IN PRACTICE

The DKIM signature is generated by the MTA (Mail Transfer Agent). It creates a unique string of characters called Hash Value. This hash value is stored in the listed domain. After receiving the email, the receiver can verify the DKIM signature using the public key registered in the DNS. It uses that key to decrypt the Hash Value in the header and recalculate the hash value from the email it received. If these two DKIM signatures are a match the MTA knows that the email has not been altered. This gives the user confirmation that the email was actually sent from the listed domain.

HOW DO DKIM RECORDS WORK?

DKIM uses two actions to verify your messages. The first action takes place on a server sending DKIM signed emails, while the second happens on a recipient server checking DKIM signatures on incoming messages. The entire process is made possible by a private/public key pair. Your private key is kept secret and safe, either on your own server or with your ESP, and the public key is added to the DNS records for your domain to broadcast it to the world to help verify your messages. Let's dive a little deeper into how DKIM works on servers that are sending and receiving email.




WHAT IS REQUIRED FOR A DKIM SIGNATURE?

DomainKeys Identified Mail (DKIM) is a digital signature added to outbound emails. It looks like a random set of characters mostly unreadable to a human user. Recipients don't see this unless they dig into the source code of the email. DKIM is meant for the recipient's email server, which authenticates the sender based on it and, if everything seems to be fine, lets pass the email to the mailbox.

Here is an example of a regular DKIM record:

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=asuswebstorage.com; s=default;  
t=1572282571; bh=NFzBvJ/pEmf+yUHDd/Y7dYNH9pE+Bx6o95KcxhwFL78=; h=From:To:Subject:From;  
b=QwgINKqwcBu3GbeWm2Be81qXks6Pq9yMmDZ19C6mT8moXVBeokpEmDN+0RyZFiOmNH30kbe6HbS21Y3b1Pf726  
UH/V/0VAH0nigTuir4TWdN/IUePV+goQdEJ2+sDQ1fHlVjyyJCRwCiFiZpBIjhTBNN0vrgNJZ/gSLL0vq6k3s=
```



It consists of the following tags:

v=1 – the version (always equals to 1)

a= – a signing algorithm used for the creation of a DKIM record

c= – a canonicalization algorithm for the header and the body

d= – a domain where the DKIM is signed

s= – a DKIM selector


t= – a timestamp of when the email was signed

bh= – a hashed email body

h= – a list of headers

b= – a digital signature

To create the DKIM signature, you will have to specify only two tags of all the above: an authorized domain (d=) and a selector (s=).



Choose a domain:

=====

When validating DKIM signatures, the recipient's mail server checks whether the domain included in the signature (d=) matches the domain included in the 'From:' field of the email.

You may use different domains for sending different types of emails, so make sure each is authorized.

NB: If you're using different domains for sending emails, you'll need to have separate DKIM signatures for each domain.

Pick a DKIM selector"

=====


A selector or a selector prefix is a name you need to specify to create the DKIM key.

During the validation process, the server runs a DNS query according to the combination of the authorized domain (d=) and the selector (s=).

This is required to fetch the public key.

Each selector is assigned to a separate private key. If you send different types of emails (marketing, transactional, etc.) from the same domain,

it's better to use separate keys for your convenience though you don't have to. You'll need to use different selectors to generate those. Pick any name for your selector.



Choose a public and private key :

=====

The domain and selector are the input data used to generate a key pair, which consists of the public and the private key. The public key is used in the DNS TXT record, whereas the private key is used for the sending MTA. Check our blog post, “What is an MTA?”, if you need to brush up on what that is.

The MTA uses the private key to hash headers (h=) and the body (bh=) of the outbound email. The private key is kept on the server and never leaves.

When an email with DKIM arrives, a receiving mail server makes a DNS query to get the public key. The server uses it to build its own hashes and then compares them with the ones received. If there’s a match, the email is let in.

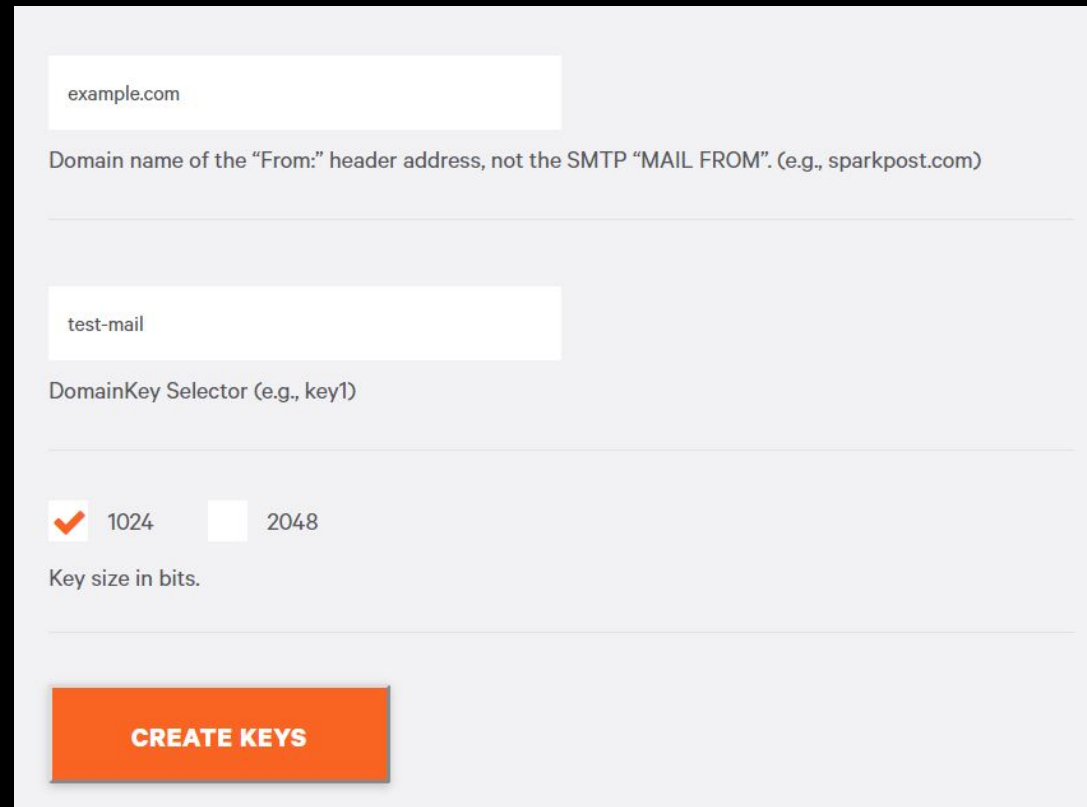
You can generate these keys with one of the following tools:

DKIM Core – the selector is assigned automatically.

DKIM Generation Wizard by SocketLabs – allows you to assign a selector and generate 1024 and 2048 bit key pairs.

GENERATE PUBLIC AND PRIVATE KEYS

Generate your public and private key pair using a dedicated tool. We're using DKIM Wizard by SparkPost, as follows:



The screenshot displays the DKIM Wizard interface with the following fields and options:

- Domain:** A text input field containing "example.com". Below it, a label reads: "Domain name of the 'From:' header address, not the SMTP 'MAIL FROM'. (e.g., sparkpost.com)".
- DomainKey Selector:** A text input field containing "test-mail". Below it, a label reads: "DomainKey Selector (e.g., key1)".
- Key size:** Two radio button options are shown: "1024" (which is selected, indicated by a red checkmark icon) and "2048". Below these options, a label reads: "Key size in bits."
- Submit Button:** An orange button at the bottom labeled "CREATE KEYS".

GENERATE PUBLIC AND PRIVATE KEYS

After the click on **Create Keys**,
we've got two keys: public and private.

-----BEGIN PUBLIC KEY-----

```
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCehqKMB6znGXo/pC83mG0bm8OW  
o4daBYBb9wqqDaf1z7Mf9KW1oaUm9j7hQq7af7jh+DSw0tXWr4HbJrI50DW/QVHq  
YK1PX3hvYUohBxg//T0u0rK30SJss3OrpkoRqd150ynYxwwLymsjIwODT7Gf9WZP  
cL86rdboSrm/ost4mwIDAQAB
```

-----END PUBLIC KEY-----

-----BEGIN RSA PRIVATE KEY-----

```
MIICXQIBAAKBgQCehqKMB6znGXo/pC83mG0bm8OWo4daBYBb9wqqDaf1z7Mf9KW1  
oaUm9j7hQq7af7jh+DSw0tXWr4HbJrI50DW/QVHqYK1PX3hvYUohBxg//T0u0rK3  
0SJss3OrpkoRqd150ynYxwwLymsjIwODT7Gf9WZPcL86rdboSrm/ost4mwIDAQAB  
AoGBAIIJibjiYUBMUjhKskzPEAQMDLS7VTdzFfqirKpYBjJJksR/hIC7Fv+7ndW6  
roYPX2ycUT4MHH64FDK1hU6JJtjfU1UxC8RGgT15lokcksUeSvecRD+/1tjAV3nF  
74UgFLXQiScTaTXnyS+hC4DsF19270hM25kx8B9bBZypeBLwBAkEA70ExCenCmK5/  
AUkjVlm1dkCDv4wQyFqMXqURWzSNk2CyoYPc30wWpDF00rmeshBNTkTje8+SY8y8  
38Izn1admWJBAKme/6lcZiwZPWzRXJc1yU3AfcInkw/yvhlbpBU01lvMi72Unh4K  
ZMPbDuwUHLXnw3i0jXq//QpWqhIPsV+ZwQECQFvK7z2WagqaKDAeF2ix0mUkK2f6  
HRZBZ0mImzga2ZaJqv880Cj70FP+hYuzm6dFieNVmtGNueSAUJaos5WwbbECQDn+  
61t1BFo/pwj17Lqm4VV8Y4NnFJl3Xhg9hTTOBe0NWxpPHmEXHKz60XdWyaifvggf  
c+BsljBcYXobc4JEiAECQQDlK+5hI60HRtS4AXWiZ91TKLc1ZxhhamEZrXuLzAR4  
88AKwQMt0mwSN/o30IvmqXATfefPnMJBMkF7YYqFwxq5
```

-----END RSA PRIVATE KEY-----

CONFIGURE THE DNS SERVER WITH THE PUBLIC KEY

Create a DKIM TXT record using the domain, selector and the public key. The record will carry the name of the authorized domain attached with the selector prefix, as follows:

`test-mail._domainkey.example.com`

The DKIM entry starts with the k= tag.

It stands for 'Key type'. Sending and receiving servers must support the rsa key type, which indicates that an ASN.1 DER-encoded public key is being used in the p= tag.

The p= tag further encodes the value using base64.

Here is what we've got:

`test-mail._domainkey.example.com IN TXT`

```
"k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCehqKMB6znGXo/
pC83mGObm8OWo4daBYBb9wqqDaflz7Mf9KW1oaUm9j7hQq7af7jh+DSw0tXWr4HbJrI50DW/
QVHqYKlPX3hvYUohBxg//T0u0rK3OSJss3OrpkoRqd150ynYxwwLymsjIwODT7Gf9WZPcL86rdboSrm/
ost4mwIDAQAB"
```

Add this DKIM entry to your domain's DNS records. In most cases, you'll have to wait 24-48 hours for the changes to take effect.

HOW CAN I TEST MY DKIM RECORD?

Feel free to use our [DKIM Inspector](#), a free diagnostic tool that you can use to test your DKIM settings if you've already started implementing DKIM for your domain(s). Our free [DKIM Validator](#) can help you verify that your DKIM record is correctly formatted.

CAN I HAVE MULTIPLE DKIM RECORDS?

A domain can have as many DKIM records for public keys as servers that send mail. Just make sure that they use different selector names.



DOMAIN-BASED MESSAGE AUTHENTICATION, REPORTING, AND CONFORMANCE,

DMARC is a DNS TXT Record that can be published for a domain to control what happens if a message fails authentication (i.e. the recipient server can't verify that the message's sender is who they say they are).

DMARC is build on top of DKIM and SPF. Together they are the best practice to prevent email spoofing and make your emails more trustworthy. DMARC only works if you have set up both SPF and DKIM. If you have proper process this carefully you can use the DMARC Analyzer tool to receive DMARC reports which contain detailed information who is sending email on your behalf.



BACKGROUND

Email authentication technologies SPF and DKIM were developed over a decade ago in order to provide greater assurance on the identity of the sender of a message. Adoption of these technologies has steadily increased but the problem of fraudulent and deceptive emails has not abated. It would seem that if senders used these technologies, then email receivers would easily be able to differentiate the fraudulent messages from the ones that properly authenticated to the domain. Unfortunately, it has not worked out that way for a number of reasons.

- Many senders have a complex email environment with many systems sending email, often including 3rd party service providers. Ensuring that every message can be authenticated using SPF or DKIM is a complex task, particularly given that these environments are in a perpetual state of flux.
- If a domain owner sends a mix of messages, some of which can be authenticated and others that can't, then email receivers are forced to discern between the legitimate messages that don't authenticate and the fraudulent messages that also don't authenticate. By nature, spam algorithms are error prone and need to constantly evolve to respond to the changing tactics of spammers. The result is that some fraudulent messages will inevitably make their way to the end user's inbox.
- Senders get very poor feedback on their mail authentication deployments. Unless messages bounce back to the sender, there is no way to determine how many legitimate messages are being sent that can't be authenticated or even the scope of the fraudulent emails that are spoofing the sender's domain. This makes troubleshooting mail authentication issues very hard, particularly in complex mail environments.
- Even if a sender has buttoned down their mail authentication infrastructure and all of their legitimate messages can be authenticated, email receivers are wary to reject unauthenticated messages because they cannot be sure that there is not some stream of legitimate messages that are going unsigned.

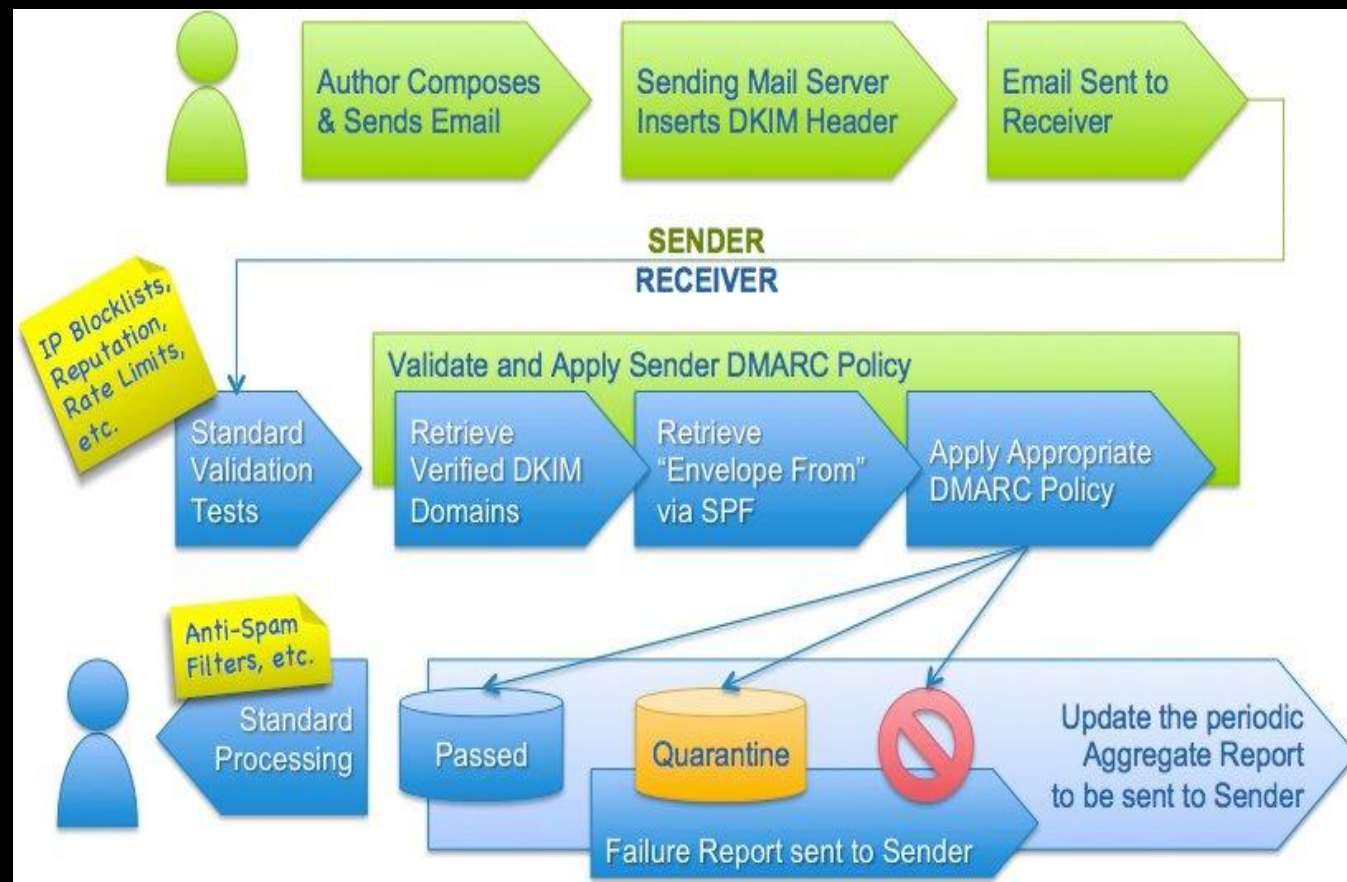
The only way these problems can be addressed is when senders and receivers share information with each other. Receivers supply senders with information about their mail authentication infrastructure while senders tell receivers what to do when a message is received that does not authenticate.

In 2007, PayPal pioneered this approach and worked out a system with Yahoo! Mail and later Gmail to collaborate in this fashion. The results were extremely effective, leading to a significant decrease in suspected fraudulent email purported to be from PayPal being accepted by these receivers.

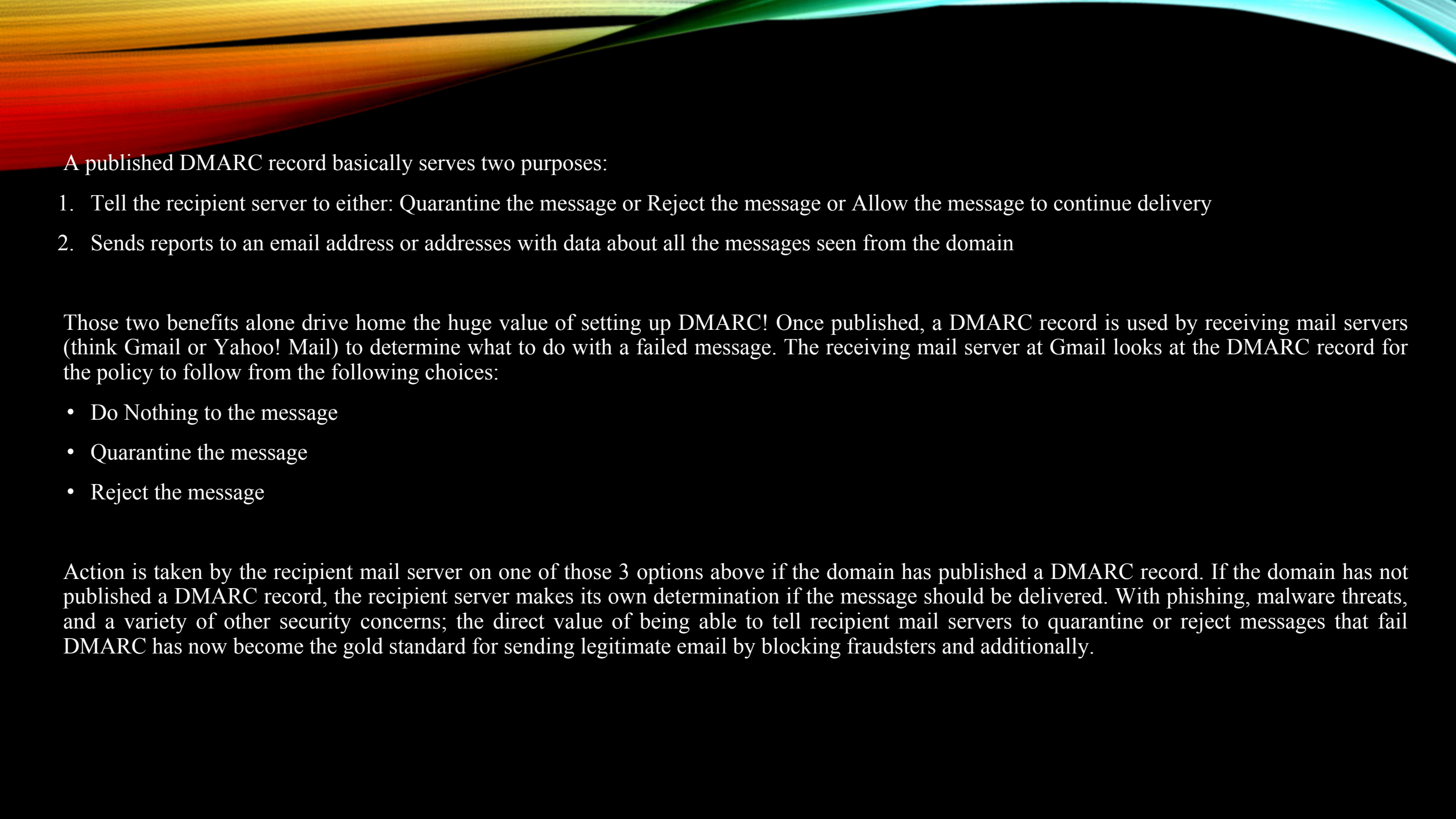
The goal of DMARC is to build on this system of senders and receivers collaborating to improve mail authentication practices of senders and enable receivers to reject unauthenticated messages.

DMARC AND THE EMAIL AUTHENTICATION PROCESS

DMARC is designed to fit into an organization's existing inbound email authentication process. The way it works is to help email receivers determine if the purported message "aligns" with what the receiver knows about the sender. If not, DMARC includes guidance on how to handle the "non-aligned" messages. For example, assuming that a receiver deploys SPF and DKIM, plus its own spam filters, the flow may look something like this:



In the above example, testing for alignment according to DMARC is applied at the same point where ADSP would be applied in the flow. All other tests remain unaffected.



A published DMARC record basically serves two purposes:

1. Tell the recipient server to either: Quarantine the message or Reject the message or Allow the message to continue delivery
2. Sends reports to an email address or addresses with data about all the messages sent from the domain

Those two benefits alone drive home the huge value of setting up DMARC! Once published, a DMARC record is used by receiving mail servers (think Gmail or Yahoo! Mail) to determine what to do with a failed message. The receiving mail server at Gmail looks at the DMARC record for the policy to follow from the following choices:

- Do Nothing to the message
- Quarantine the message
- Reject the message

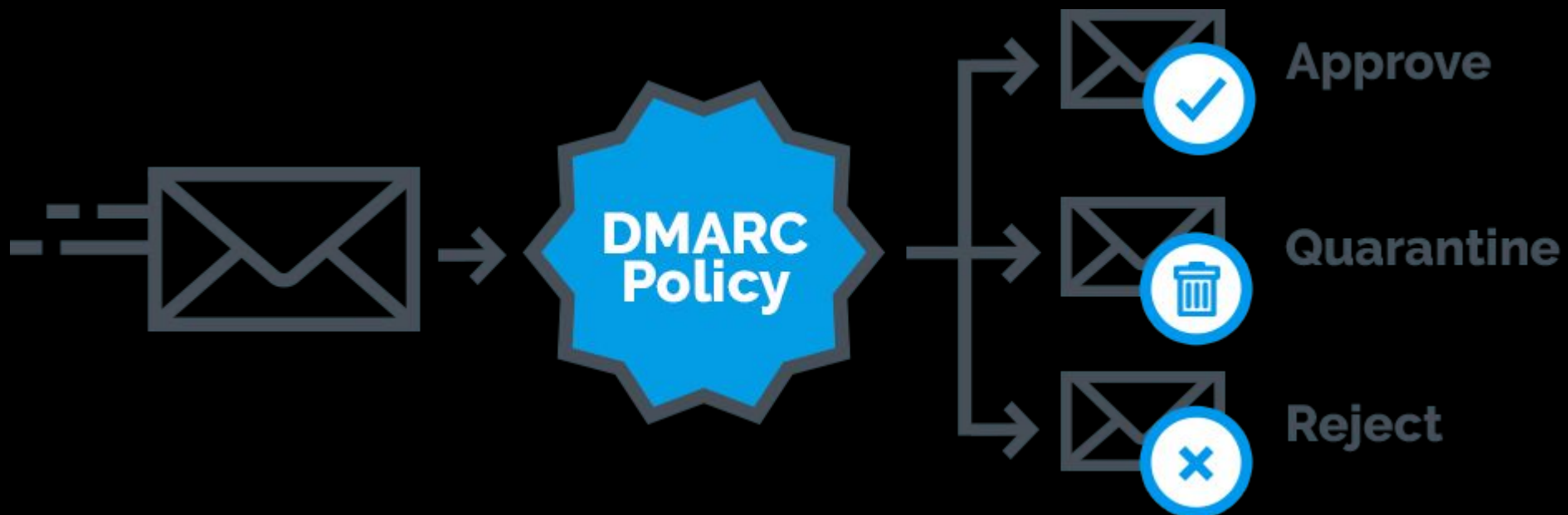
Action is taken by the recipient mail server on one of those 3 options above if the domain has published a DMARC record. If the domain has not published a DMARC record, the recipient server makes its own determination if the message should be delivered. With phishing, malware threats, and a variety of other security concerns; the direct value of being able to tell recipient mail servers to quarantine or reject messages that fail DMARC has now become the gold standard for sending legitimate email by blocking fraudsters and additionally.

DMARC POLICIES

When you're using DMARC you can set up a policy to define how you want the receivers to handle emails which fail the DMARC checks.

You can choose one of these 3 DMARC policies:

- **none:** Just monitor the results and do not take specific action for failing messages. Use this policy to start gathering DMARC reports and analyzing the data in these reports.
- **quarantine:** Put the messages which fail the DMARC checks in quarantine. This mostly means that receivers will place these messages in the junk folder.
- **reject:** Reject all messages which fail the DMARC checks. The receivers should do this 'on SMTP level' which means the messages will bounce directly in the sending process.



ANATOMY OF A DMARC RESOURCE RECORD IN THE DNS

DMARC policies are published in the DNS as text (TXT) resource records (RR) and announce what an email receiver should do with non-aligned mail it receives.

Consider an example DMARC TXT RR for the domain “sender.dmarcdomain.com” that reads:

`"v=DMARC1;p=reject;pct=100;rua=mailto:postmaster@dmarcdomain.com"`

In this example, the sender requests that the receiver outright reject all non-aligned messages and send a report, in a specified aggregate format, about the rejections to a specified address. If the sender was testing its configuration, it could replace “reject” with “quarantine” which would tell the receiver they shouldn’t necessarily reject the message, but consider quarantining it.

DMARC records follow the extensible “tag-value” syntax for DNS-based key records defined in DKIM. The following chart illustrates some of the available tags:

Tag Name	Purpose	Sample
v	Protocol version	v=DMARC1
pct	Percentage of messages subjected to filtering	pct=20
ruf	Reporting URI for forensic reports	ruf=mailto:authfail@example.com
rua	Reporting URI of aggregate reports	rua=mailto:aggrep@example.com
p	Policy for organizational domain	p=quarantine
sp	Policy for subdomains of the OD	sp=reject
adkim	Alignment mode for DKIM	adkim=s
aspf	Alignment mode for SPF	aspf=r

HOW SENDERS DEPLOY DMARC IN 5-EASY STEPS

DMARC has been designed based on real-world experience by some of the world's largest email senders and receivers deploying SPF and DKIM. The specification takes into account the fact that it is nearly impossible for an organization to flip a switch to production. There are a number of built-in methods for “throttling” the DMARC processing so that all parties can ease into full deployment over time.

1. Deploy DKIM & SPF. You have to cover the basics, first.
2. Ensure that your mailers are correctly aligning the appropriate identifiers.
3. Publish a DMARC record with the “none” flag set for the policies, which requests data reports.
4. Analyze the data and modify your mail streams as appropriate.
5. Modify your DMARC policy flags from “none” to “quarantine” to “reject” as you gain experience.

DMARC RECORD GENERATOR

The DMARC Record Wizard allows you to create your DMARC Record ready for publication for your domain, so that you're able to gain valuable insights on who is abusing your domain.

Not sure what a DMARC record is? Read more about it [here](#).

Our Wizard guides you through each step of the process, including explanation.

- Step 1: Enter the domain
- Step 2: Choose your Policy
- Step 3: Provide your Aggregate reports address
- Step 4: (Optional) Provide your Failure Reporting address
- Step 5: Choose Identifier Alignment
- Step 6: (Optional) Choose Subdomain Policy
- Step 7: (Optional) Choose DMARC Policy percentage

<https://dmarcian.com/dmarc-record-wizard/>