How to implement DMARC - Step by Step

This section serves as a mini-guide with the goal to help you jump right in implementing DMARC. These actionable steps outlined below will help you avoid being overwhelmed by a myriad of technical acronyms and maintain your sanity. DMARC is not hard after all.
If you'd like to learn more about SPF/DKIM/DMARC and email overall, feel free to start from the beginning of this guide.
Now let's implement DMARC. First things first. Inventory your email domains you want to protect, and do the following steps for each one of them.

Step 1: set up SPF

Since DMARC is based on SPF and DKIM, it's a good idea to start from the basics. Now let's set up SPF.

Create an SPF record, then publish the SPF record.

Once you have saved the settings in your DNS console, give it up to 1 hour for the settings to propagate across the DNS.

Check the SPF record to make sure SPF is set up correctly before moving to the next step.

Step 2: set up DKIM

Next set up DKIM.

Create a DKIM record, then publish the DKIM record.

If there are multiple DKIM records, make sure to publish all of them.
Similar to the SPF setup, after you have saved the settings in your DNS console, give DNS propagation up to 1 hour.

Check the DKIM records to make sure DKIM is set up correctly before moving to the next step.

Step 3: publish a DMARC record

Now that SPF and DKIM are ready, it's time to set up DMARC.

Create a DMARC record, then publish the DMARC record.

Remember to set the DMARC policy to none to start in monitoring mode, so that no legitimate email message will be negatively affected.

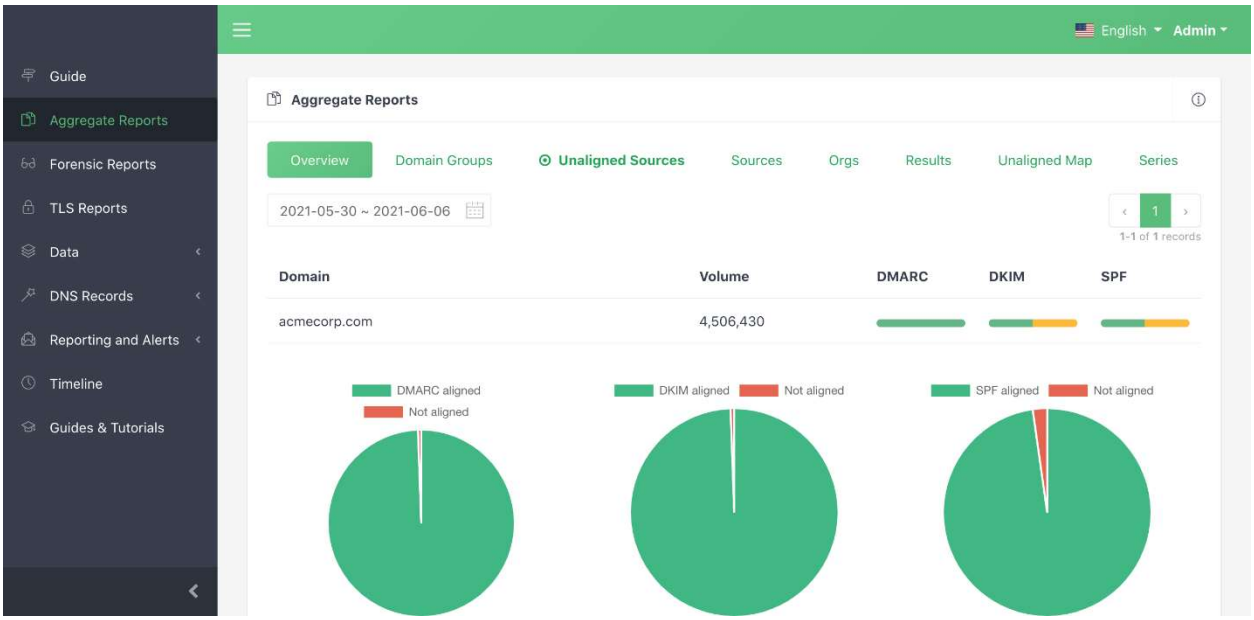
Check the DMARC record to make sure the DMARC record is correctly published after ~1 hour.
After this is done, remember DMARC aggregate reports are usually sent daily, therefore, it might take 1 day or 2 before the reports land in your designated mailbox (specified by the `rua` tag in your DMARC record).

Step 4: analyze DMARC reports

Now the reports are ready! The situation can be one of:

- you use a DMARC analytics service like DMARCLY to handle all the report parsing/rendering for you; or
- you specify your own mailboxes to receive the reports.

In the first scenario, you don't have to do the chores: setting up mailboxes, downloading reports, parsing, rendering, etc. Everything is done for you. All you have to do is to log in to the dashboard, view and analyze the charts. Here is an example:



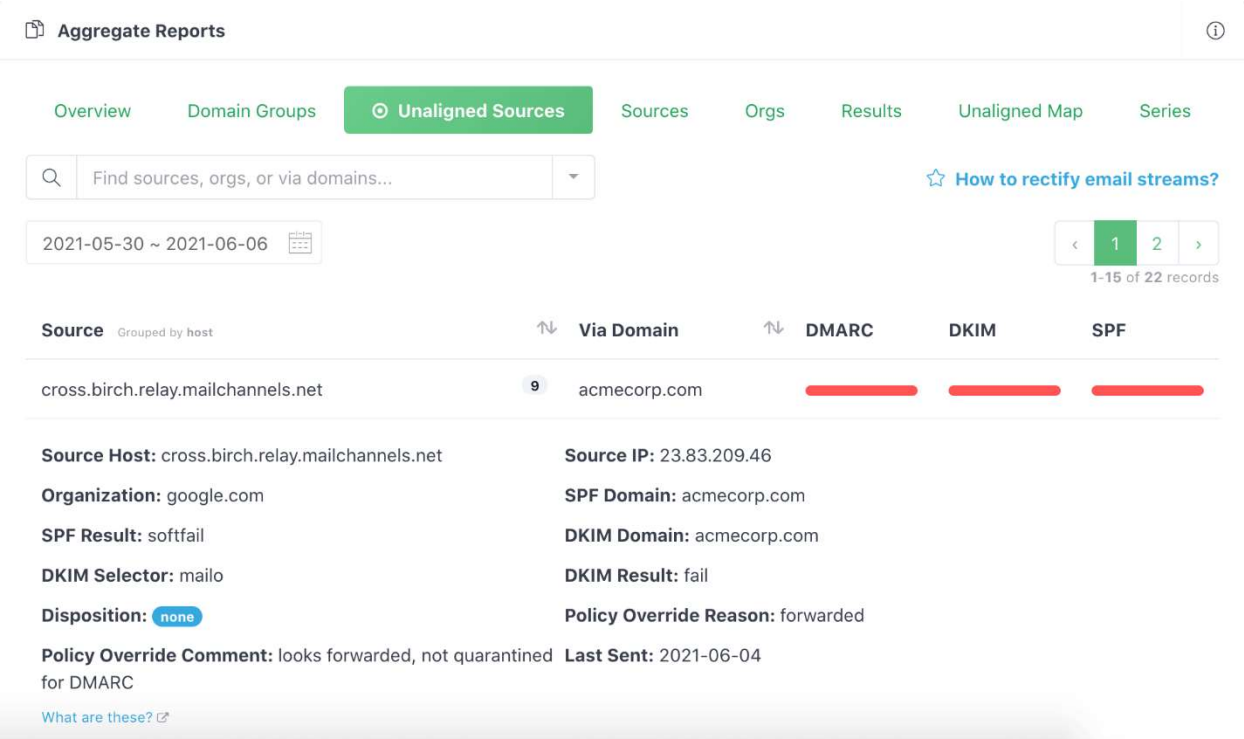
In the second scenario, you need to perform the tasks mentioned above manually for each report.

Step 5: rectify email streams

Rectifying an email stream means setting up SPF and DKIM for the source of the stream, so that all emails from that source pass DMARC authentication.

You need to identify all unauthenticated legitimate email sources, add them to the email streams, and remove everything else.

The dashboard allows you to do this easily. Log in and go to Aggregate Reports, then click the Unaligned Sources tab:



You will need to go through the unaligned sources, and add all of the legitimate ones to your email streams, so that emails from such sources pass next time.
Here is a post on how to set up SPF and DKIM for Mailchimp. If you are using other email delivery services, the steps are similar; however, the details of setting up SPF and DKIM are different, therefore you should look up the documentation of the particular service.
Next keep checking your email streams. If you find any additional unauthenticated legitimate email sources, rectify them as described above.
If you find emails from all legitimate sources pass DMARC authentication, and everything else fails, say for ~1 month, consider moving to the quarantine mode.

Step 6: transition to quarantine mode

In the quarantine mode (`p=quarantine`), any email message that fails DMARC authentication is moved to the spam folder. You can consider this mode somewhat "between" no action at all (`p=none`) and outright rejection (`p=reject`).
If you find everything fine for ~3 months, consider moving to the reject mode.

Step 7: transition to reject mode (full DMARC implementation)

In the reject mode, any email message that fails DMARC authentication is rejected outright. It's the harshest action taken on emails that fail authentication and provides complete email protection against spoofing.

Step 8: ongoing monitoring and updating

You have gone through a full cycle of DMARC implementation. But it's not the end of the story. From now on you need to keep monitoring your email streams, just to make sure everything stays fine within your email program. As the email infrastructure within your organization changes from time to time, what worked in the past might not work now. When this happens, you need to analyze the reports and update your setups accordingly, so that your domain maintains high email deliverability and a good sender reputation.

An end-to-end SPF/DKIM/DMARC wizard

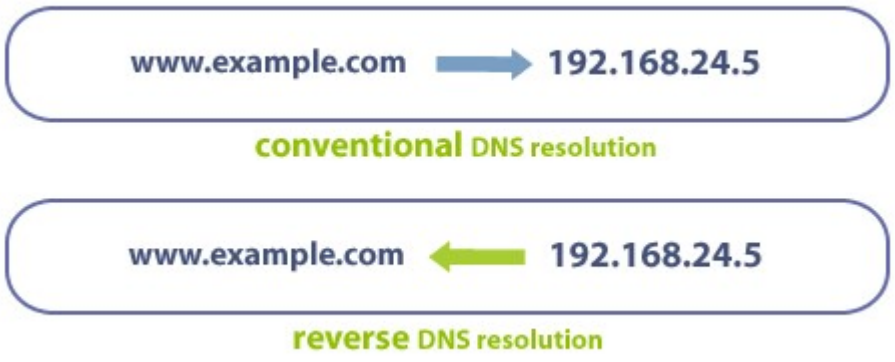
What's even greater than the above very actionable steps, I have implemented an end-to-end SPF/DKIM/DMARC wizard. This wizard will tour your through every step toward a complete email authentication deployment, including SPF, DKIM, and DMARC.
You can take it for a spin here: End-to-end SPF/DKIM/DMARC wizard.

Other considerations

In addition to the steps outlined above, to accomplish a complete email authentication implementation for your organization based off of SPF, DKIM, and DMARC, you also need to take the following scenarios into account, if applicable.

Setting up reverse DNS records (PTR records)

If you send emails from your on-premises server, you must set up the reverse DNS record, otherwise many ESP's will block your email.
A reverse DNS record, as the name suggests, maps an IP address into a host name, exactly the reverse of what a normal DNS A record does.
Here is an illustration of how a reverse DNS record works VS how a normal DNS A record works:



Best practices

Any IP address you send email from should have a reverse DNS record (PTR record) which allows you to take a numeric IP address and get a hostname. Many ESPs will block your emails, if no PTR record is set up. This also enables you to view email sources as hostnames, rather than IP addresses.

Implement all three of SPF, DKIM, and DMARC. That's because authenticated email makes it easier to build and monitor the reputation of the sender. Good senders using authentication can be more easily recognized, while those not using it will increasingly start at a disadvantage.

SPF best practices

SPF has a lot of complex and powerful mechanisms, but don't take that as a challenge to find ways to use them. Keep your SPF records as simple as possible, and don't put any more hosts in your SPF records than you have to. This applies to the include: mechanism as well – use as few as possible, avoid nested includes whenever you can, and never use so many includes that you go over the 10 lookup limit.

If you specify blocks of addresses using CIDR notation in your SPF records, only use ranges between /30 and /16 inclusive (example: 10.10.10.0/24) – the higher the number after the slash, meaning a smaller block of addresses, the better. Avoid anything in the range /1 to /15 because some receivers will discount such blocks or even ignore them completely. And never, ever use or include a record with "+all" in it. The only way to productively use "all" is in the "~all" or "-all" mechanisms.

DKIM best practices

First, make sure your DKIM keys are at least 1,024 bits long. Signatures made using keys shorter than 1024 bits will often be ignored completely, and this practice will become more widespread as more senders switch to keys of 2,048 bits or longer.

However, let's not go to the other extreme either. You should not use 4096 bit or longer keys, as these may not fit within a 512-byte DNS UDP response packet. Since some DNS implementations don't allow packets larger than 512 bytes, using super long keys might cause DKIM to fail.

DKIM is built on cryptographic digital signatures, and the science of cryptology has developed its own best practices over time. One of these is to change cryptographic keys regularly, so that bad actors don't have years and years to try to attack the key. Unfortunately, many email senders are using keys that were created five years ago, and sometimes longer! Instead, you should switch to a new DKIM key, or "rotate" your keys, at least once a year. If you send millions of

messages each month, or if they are particularly sensitive messages, you should really consider rotating your keys more often than that.

DMARC best practices

Start using DMARC in `p=none` first. You'll benefit from the reporting it provides, and it allows you to signal to email service providers that they should identify messages using your domain that don't pass authentication.

Then gradually progress to `p=quarantine` and ultimately to `p=reject`. This means fraudulent messages using your domain can be identified and blocked, which will help protect your employees and customers while improving your domain's reputation.

Parsing raw DMARC reports is time-consuming and error-prone. Outsourcing DMARC report analytics to a modern DMARC report analyzer service like dmarcly.com can save a lot of time and help you reach `p=reject` faster.

Final words

Thanks for staying with me all the way. It's a long read.

I hope this comprehensive DMARC implementation guide has shown the importance of email authentication for modern organizations, explained the concepts well, and provided actionable steps to an effective SPF/DKIM/DMARC implementation.

If you have any questions about how to set up DMARC/DKIM/SPF, or just anything related to email authentication, feel free to reach out at `support@mail.dmarcly.com`. I'll be happy to answer!

Good luck implementing DMARC, DKIM, and SPF within your organization, and goodbye to email spoofing!