Email Security with SPF, DKIM, DMARC

CySecurus Infosec Private Limited
connect@cysecurus.com

CySecurus

Email Security with SPF, DKIM and DMARC

Sender Policy Framework (SPF)

Sender Policy Framework (SPF) is an email authentication technique used to prevent spammers from sending messages on behalf of your domain. With SPF, an organization can publish details of your authorized mail servers.

SPF is a DNS text entry that shows a list of servers allowed to send mail for a specific domain. The domain owners/administrators are the only people allowed to add/change the main domain zone. Hence SPF in a DNS entry can be considered a way to enforce the fact that the list is authoritative.

Steps to implement SPF.

1. Collect the IP addresses or address ranges of the servers that are used to send emails. Emails can be sent from webservers after a user fills a form, they can be sent from the email servers, they can be sent from applications like Tally, SAP, Payroll, monitoring systems, alert systems, and other such systems.
2. Collect IP addresses or address ranges or the SPF domain of the third-party email vendor who sends emails like marketing emails on your behalf. They can be companies like Mailchimp, sendinblue etc.
3. Create your SPF Record.
    a. Start with the SPF version, this part defines the record as SPF. An SPF record should always start with the version number v=spf1 (version 1) this tag defines the record as SPF. There used to be a second version of SPF (called: SenderID), but this was discontinued.
    b. After including the v=spf1 SPF version tag, you should follow with all IP addresses that are authorized to send an email on your behalf. For example: v=spf1 ip4:34.243.61.237 ip6:2a05:d018:e3:8c00:bb71:dea8:8b83:851e
    c. Next, you can include an include tag for every third-party organization that is used to send an email on your behalf, e.g. include:thirdpartydomain.com. This tag indicates that this particular third party is authorized to send an

email on behalf of your domain. You need to consult with the third party to learn which domain to use as a value for the 'include' statement.

d. Once you have implemented all IP addresses and include tags, you should end your record with an ~all or -all tag. The all tag is an important part of the SPF record. It indicates what policy should be applied when ISPs detect a server that is not listed in your SPF record. Suppose an unauthorized server does send email on behalf of your domain. In that case, action is taken according to the policy that has been published (e.g. reject the email or mark it as spam). What is the difference between these tags? You need to instruct how strict servers need to treat the emails. The ~all tag indicates a soft fail, and the -all indicates a hardfail. The all tag has the following basic markers: *-all Fail* – servers that aren't listed in the SPF record are not authorized to send an email (not compliant emails will be rejected). *~all Softfail* – If the email is received from a server that isn't listed, the email will be marked as a soft fail (emails will be accepted but marked). *+all* We strongly recommend not to use this option, this tag allows any server to send email from your domain.

There are many available SPF tags, more information can be found at the SPF parts explanation page here https://www.dmarcanalyzer.com/spf/spf-record/

• 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, it is recommended 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.

4. Publish SPF record in the DNS.

Domain Keys Identified Mail (DKIM)

Domain Keys Identified Mail (DKIM) is an email authentication technique that allows the receiver to check that an email was indeed sent and authorized by the owner of that domain. This authentication is done by giving the email a digital signature. The recipients can know if the message has not been faked or altered in transit.

Here are the steps to setup DKIM for an on-prem server:

1. List all your sending domains
2. Install a DKIM package on your email server
    a. You'll need to install a DKIM package, like OpenDKIM, on your email server. Your choice of DKIM package will depend on the email server's operating system. The installation process will depend on the DKIM package and operating system.
3. Create the public and private DKIM key pair
    a. Use a DKIM key wizard to generate a public and private key pair. You can find many like this one by simply Googling "DKIM wizard."
    b. You'll have to specify selector names for your key pairs. Selectors tell receiving email servers where to find the public key for each domain. It's best to make selectors descriptive of what their domain sends. For example, the selector for your email marketing domain could be "marketing."
4. Publish the public DKIM key
    a. Your DKIM wizard should return a selector record that should look something like this: **(selector)._domainkey**
    b. You'll need to add a TXT record with that name to your DNS. The value of the record is a specially-formatted version of your DKIM key and some identifying information that tells receivers how to interpret your DKIM key. The complete record will look something like this, which is the DKIM record for yourdomain.com: **s1024._domainkey.yourdomain.com. v=DKIM1; k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDQw PqBxkIOc1YVnJv3Occfbd3S68**
5. Hide the private DKIM key

    a. Your DKIM wizard should also produce your private key, which should be stored wherever your DKIM package specifies.
6. Configure your email server
7. Test your DKIM setup
    a. Use a DKIM record checker to make sure receiving email servers can locate your public key. They'll look for this when they're verifying that an email came from your domain.

Domain-based Message Authentication, Reporting & Conformance (DMARC)

DMARC, which stands for "Domain-based Message Authentication, Reporting & Conformance", is an email authentication, policy, and reporting protocol. It builds on the widely deployed SPF and DKIM protocols, adding linkage to the author ("From:") domain name, published policies for recipient handling of authentication failures, and reporting from receivers to senders, to improve and monitor the protection of the domain from fraudulent email.

Steps to implement DMARC

1. After implementing SPF and DKIM, check if they are signing correctly.
2. Tie up with a provider like Kratikal, Rediff.com, ProDmarc, Valimail etc., for DMARC Analyzer.
3. Get the email address to send the RUF, and RUA reports from the providers. (Note: Not all providers give RUF email address)
4. Generate DMARC record with the above RUF and RUA email addresses. Initially, the policy should be set as none. The record should look like this: v=DMARC1; p=none; fo=1; rua=mailto:dmarc_agg@auth.example.com;ruf=mailto:dmarc_afrf@auth.example.com
5. Publish this DMARC record in the DNS. The host value in the DNS record should be: _dmarc
6. Analyze the feedback you receive from the DMARC Analyzer and adjust your mail streams as needed. Suppose unqualified mail gets sent to, and received by, recipients participating in DMARC. In that case, the recipient will generate reports for these messages and send them back to the mailto: address specified in your DMARC record. These reports will give you the information required to evaluate and tune your mail streams, helping you determine exactly what services are sending mail on behalf of your domain.
7. Escalate your DMARC policy tags from **p=none** to **p=quarantine** to **p=reject** as you gain experience
    a. Until now, you should have been using the p=none policy to get reports of any errant behaviour, and you should have a good idea of where email is coming from. The next step is to adjust the policy on your DMARC record to

start controlling how receivers handle email claiming to be from your domain.

b. **p=none** - Get reports of infractions, but no action is taken by recipients to process the messages themselves.
c. **p=quarantine** - Unqualified mail goes directly to spam but can be recovered. This is useful when you're pretty sure you know all the locations where mail is coming from but want to "softfail" any messages that are unqualified until you're 100% sure.
d. **p=reject** - The recipient mail server completely deletes unqualified mail, never to be seen again. Use this setting when you're absolutely sure you know every server and service that is sending email for your domain, signing is in place for each of these services. You want any service claiming otherwise to be denied entirely.

References

https://tools.ietf.org/html/rfc7208
https://tools.ietf.org/html/rfc6376
https://tools.ietf.org/html/rfc7489
https://mxtoolbox.com/dmarc/spf/spf-record-tags
https://blog.mxtoolbox.com/2020/11/28/dkim-signature-tags-a-primer/
https://mxtoolbox.com/dmarc/details/dmarc-tags
https://mxtoolbox.com/dmarc/dkim/dkim-alignment
https://mxtoolbox.com/dmarc/dmarc-email-tools?referrer=cms_dmarchome