DKIM Setup Guide

Well here we are, we have made it to our final setup guide DKIM. Arguably this is the most complicated email authentication protocol out there. By using cryptographic authentication DomainKeys Identified Mail (DKIM) allows an organization to take responsibility for transmitting a message so that it can be verified by mailbox providers. We are going to break down what is needed for DKIM and then walk you through creating your own policy.

Identifying what message elements to sign with DKIM

So as a sender you will decide what elements of the email are to be included in the DKIM signing process. You can include the entire message (both header and body) or just target one or more fields of the email header. The elements you decide to include in the DKIM signing process must remain unchanged in transit, or the DKIM signature authentication will fail. For example, let's say you have an email from Yahoo that you need to forward to a Google account, Yahoo may add a line of text to the top for the email (e.g. "forwarded by Yahoo mail"). So, by Yahoo adding this it has changed the body of your original email, and if you included the body element in your DKIM policy the authentication for the forwarded email will fail. However, if you include only elements from the header, say the "from" field into your DKIM signature and the message was then forwarded from Yahoo to Gmail, then DKIM authentication would pass since the header information wasn't changed.

The encryption process

Let's talk about what the signing process looks like. Cryptography is at the core of DKIM. The Sender will configure their email platform to automatically take the elements they want signed and create a hash. This process converts readable text into a unique textual string. For instance you have selected to only include elements from the header field "from" and "subject" lines. Using the MD5 hashing process it would look like this:

From: Thomas Anderson Thomas.anderson@yourorganization.com

Subject: The One

Maps to the following unique hash string:

c50e98bb3f15c3f055bba1cf5b64a46f

Before the email is sent the hash string is encrypted using a private key. The private key is assigned to a combination of domain and selector, thus allowing you to have multiple private keys for the same domain. Only the sender will have access to the private key. Once the encryption process is complete, the email is sent.

Validating the DKIM signature with a public key

The DKIM signature tells the receiving email server the domain/selector combination used during the encryption process. In order to validate this the email server will run a DNS search to find the corresponding public key for that domain/selector combination. The public key is uniquely created to match only the private key of that specific domain/selector combination, this is also known as a keypair match. By using the keypair match the email server is then able to decrypt the DKIM signature to the original hash string. Once the email server has the original hash, it will take the elements signed by DKIM and generates its own hash. Finally, the email server will compare the decrypted original hash with the newly generated hash from the DKIM signature. If these match, we can conclude that

- The DKIM domain really does own the email
- The elements of the email signed by DKIM were not changed in transit

Configuring DKIM on mail servers

Ok now that we understand how DKIM operates, let's work on getting it setup on your mail servers. There is something I want to take a minute and stress. DKIM deals with public and private key authentication it is imperative that you only publish your public key and store your private key in the correct directory on the server. I know this may seem like the most straight forward part I mean after all they are labeled and everything. But a simple oversight and you could publish/store the wrong key and then have a wild goose chase on your hands after you figure out that DKIM is failing. Everything will look correct but that one little crossed wire will cause a major headache. Know back to our regularly scheduled setup guide, as with the SPF setup guide and DMARC setup guide you want to start with:

- Inventory all of your sending domains
    - With DKIM these will need to be more in depth and should include any vendors that deploy emails for
        - Marketing
        - Customer service
        - Corporate
    - Also contact department heads
        - Client services
        - Internal IT
        - Email administrator
        - Email service provider
- Install and configure DKIM on your mail server
    - With all outgoing mail needing the DKIM signature we need to install a DKIM package directly to the server
    - If using a mail provider, they will help setup the DKIM record
- Create your public and private keypair
    - There are wizards that can help with this
        - https://dkimcore.org/tools/
        - https://port25.com/dkim-wizard/
        - Or you can create your own using openssl

Here is what part of a public key will look like:

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

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAhO+AQr1/U82LIaEKl8
ztnjQfikJtcvkoqgsFH3MINHVnP0rTgrr4FHiwjR3twFzebHnDVnAdxMUdLs5CwHa
+CwWqh4qpz+9o90HNejnACcjQvdnw3lky5Sn5sL5osqvgctg2nCTHUz3y7tym7o
3H
```

Here is what part of a private key will look like:

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

MIIEpAIBAAKCAQEAhO+AQr1/U82LIaEKl8ztnjQfikJtcvkoqgsFH3MINHVnP0rTgr
r4FHiwjR3twFzebHnDVnAdxMUdLs5CwHa+CwWqh4qpz+9o90HNejnACcjQvdn
w3lky5Sn5sL5osqvgctg2nCTHUz3y7tym7o3HWiUDsYxPi68gIF6m8bhqH8Umbk
mv
```

    - First you enter the From: domain that you are authenticating
    - Enter your selector name
        - We recommend being descriptive to the type of email being sent
            - Department names
            - Newsletter
    - Ensure your key is encrypted with at least 1024-bit encryption or higher
- Publish your public key
    - The DKIM record includes the subdomain which is a combination of the domain and selector name, which will look something like this:
        - Selector._domainkey.yourorganization.com
    - Your public key will be stored in the TXT portion of that domain
- Store your private key
    - The private key needs to be stored in the DKIM directory on your organizations email server

Problems with DKIM

As we have seen DKIM is a great tool to help secure your organizations email, but it does have some flaws. DKIM is vulnerable to what is called a “replay attack”. This type of attack is inherently a store-and-forward protocol. Since DKIM has no sender to recipient handshake it will not confirm when a keypair is used more than once. The attackers will save a previously authorized email and add a secondary “From” field to the header that the end user will likely never notice. As this does not change the original hash, DKIM will allow this to pass and be sent to the recipient mailbox. There is good news though, DKIM does work hard to mitigate these types of attacks. One of the major pluses that DKIM has is that it does stop any attacker from changing the information to an already signed email. If the attacker changes any elements that you have already set DKIM to hash and sign the email will fail validation. Another way to help mitigate this would be to secure your servers not with just DKIM, but with the trifecta SPF, DKIM, and DMARC.