SMTP vs LDAP Authentication

Environment

Netm**ail** Secure (all versions), **M+**Guardian (all versions)

Synopsis

The "authentication" settings in NMS (the authentication portion of a "mail route" policy) serves 2 purposes,

1. to validate a user's credentials when they log in to access the quarantine.
2. to determine if a user is valid (if the user exists) before NMS will accept mail for the user.

There are multiple types of authentication that can be used in NMS but by far the 2 most common are, SMTP and LDAP. For the first purpose (validating credentials) there is no real advantage in using LDAP vs SMTP auth but for the second purpose (validating a user's existence) there are several reasons why we suggest using LDAP authentication. To understand the benefits, let's first make sure we understand what happens during user validation...

A server out there needs to deliver a message to user@yourdomain.com, the sending server will connect to the SMTP service on your NMS and start the SMTP transaction,

> lines that begin with ">" represent the commands sent by the sending server to your NMS server.
>
> lines that begin with "<" represent the responses sent from your NMS server to the sending server.

```
< 220 yourdomain.com Netmail Secure Extreme Email Engine
> helo sending_server_name
< 250 yourdomain.com Pleased to meet you
> mail from: <user@senderdomain.com>
< 250 Sender OK
> rcpt to: <user@yourdomain.com>
...
```

Let's pause here for a moment. Notice that for every command sent by the sending server the receiving server (NMS) returns a response. The 3-digit prefix included in the SMTP responses have specific meaning,

> 2xx = positive/success
>
> 4xx = temp error, the sending server should retry later (based on the sending server's retry schedule)
>
> 5xx = permanent error, the sending server should give up (usually alerting the sender of the failed delivery via a "DSN", Delivery Status Notification aka bounce.

Before the SMTP transaction can proceed, in order for NMS to respond to the sending server's "rcpt to" command, NMS needs to determine if the recipient exists or not (sometimes referred to as "lookahead"), how the user validation is performed depends on the type of authentication you have configured in NMS.

SMTP Authentication,
The NMS will connect to the SMTP server defined in your authentication settings which we often refer to as the "backend" server (if you use GroupWise this would be your GWIA, if you use Exchange this would be the HubTransport) only this time NMS is the "sending" server, it will initiate an SMTP conversation like we see above until it reaches a point where it can "rcpt to" the same recipient. If the backend server returns a 2xx response (250 Recipient OK) it means it accepts mail for the recipient, if the backend server returns a 5xx response (550 Mailbox not found) it means it does not accept mail for the recipient.

LDAP authentication,
The NMS server will query the directory service you have defined in your authentication settings (if you use GroupWise this would be an eDirectory server, if you use Exchange this would be an Active Directory server), using an ldap search filter crafted from the settings in your authentication settings to look up/find the recipient user object. If LDAP query finds a matching object, we know the recipient is valid, if no matching objects are found, the recipient is not valid.

Which ever method is used to validate the recipient (SMTP or LDAP) NMS has now determined if the recipient exists or not and can provide a response to the sending server's "rcpt to" command, usually either,

```
> 250 Recipient OK
or
> 550 Mailbox not found
```

If the recipient is "OK" the sending server can proceed and give NMS the message it is attempting to deliver. While it may have taken several minutes to reach this point in this article, the recipient validation process usually takes less than a second.

Now that we understand how we validate recipients and that we will only accept mail for valid recipients, we can move on to explaining why we consider LDAP to be a superior method to validate users than doing so via SMTP.

Most enterprise class email systems (Exchange, GroupWise, etc.) store user information in a directory in the form of user objects, a user object has various attributes such as a username, first name and last name, etc. Typically username@yourdomain.com is the primary/default email address for the user but the mail system will usually allow for other common email address formats based on the user's first and last name or other custom email address formats. For example, the following email addresses might all be valid for the same user,

first.last@example.com
flast@example.com
nickname@example.com

If we validate users via SMTP, mail can be sent to any of those valid email addresses, and for each one NMS will query the backend and get a response of "250 Recipient OK", so we can determine that they are all valid email addresses but what we cannot determine is that they all belong to the same user, the response we get is limited to a "yes" or "no" which means that NMS would populate the 3 valid email addresses as 3 separate users. This is not optimal for the following reasons,

- NMS will gobble up 3 user license for the same user. Now we don't expect you to purchase 3 times the licenses you actually require, if you find yourself in this situation we can make a license exception but this is extra work for us.
- Policy assignment at the user level becomes more difficult to manage, this is extra work for you, the admin.
- A separate quarantine mailbox exists for each user which means the user might receive multiple quarantine reports and requires extra work to manage his/her quarantine.

Now behold, the beauty of validating users via LDAP!
Here is an ldif representation of what a user object typically looks like on the backend system (AD/eDir), only relevant attributes are shown,

```
dn: cn=danielb,ou=users,o=example
objectclass: Person
name: Daniel
sn: Bigras
mail: danielb@example.com
mail: daniel.bigras@example.com
mail: dbigras@example.com
mail: dan@example.com
cn: danielb
```

The attributes might be named differently depending on the type of directory service used by the backend mail system but regardless of the attribute names, user objects in all the directory services I have encountered usually have the following 2 things in common,

1. a non unique attribute exists for all valid email addresses, in the above example the "mail" attribute.
2. a unique attribute contains the username, in the above example the "cn".

When looking up the user via LDAP we search for an object using the non unique attribute and if a match is found we ask that the unique attribute be returned. This means,

- a message comes in for dan@example.com, we search for an object where the value of the "mail" attribute contains dan@example.com, if we get a match the value of the "cn" attribute is returned, thus "danielb".
- a message comes in for daniel.bigras@example.com, we search for an object where the value of the "mail" attribute contains daniel.bigras@example.com, if we get a match the value of the "cn" attribute is returned, again "danielb".
- a message comes in for danielb@example.com, we search for an object where the value of the "mail" attribute contains danielb@example.com, if we get a match the value of the "cn" attribute is returned, again "danielb".

See what is going on here? Regardless which of the user's valid email addresses we search for the same unique username is returned, "danielb". Unlike validating users via SMTP where all we know is that the email address is valid or not, via LDAP we know that the email address is valid and also that it belongs to the same user! This way we don't run into the same problems we had when validating users via SMTP, we don't populate multiple user objects in NMS for each of the backend user's valid email addresses, we know they all belong to the same user. No usercount/license bloat, no extra management effort for the admin or the users in managing multiple user objects or quarantine mailbox!

And that, my friends, is why validating users via LDAP is more desirable than doing so via SMTP.

Solution

N/A

Notes

Help us improve!	
Is this article helpful?	Yes No
Is it well written?	Yes No
Is the content complete?	Yes No