Avoiding email spoofing with Sender Policy Framework (SPF)

In a previous article, we have seen an external attacker could send an email carrying an address belonging to your domain, and how to detect it via the headers. **Sender Policy Framework (SPF)** is an email authentication protocol designed to prevent this kind of email spoofing by allowing domain owners to specify which mail servers are authorized to send emails on behalf of their domain (more exactly with an envelope header MAIL FROM: containing his domain).

SPF helps validating the real source of the email, reducing the likelihood of malicious actors sending emails that appear to come from a legitimate source.
Note: SPF validates the *envelope* MAIL FROM: header, **NOT** the From: email header, which is usually displayed by your email client.

How SPF works

If the mail administrator of the *example.com* domain wants to use SPF to avoid external users sending mails pretending they are coming from his mail servers, he needs to create and publish a DNS TXT SPF record, listing the identification of the servers that are allowed to send email on behalf of his domain.

Then, he and other mail administrators of domains receiving emails from the *example.com* domain need to enable SPF checking so the email handling software can alert on potential spoofing.

Note: if a mail server is not configured to check SPF, no defence against spoofing will be put in place even if the sender has an SPF record.

Let's assume *user1@example.com* sends a mail to *user2@otherdomain.com*. When SPF checking is enabled, the following checks happen:

- the receiver server extracts the domain part of the MAIL FROM: envelope header (in our case: *example.com*);
- the receiving server (in the *otherdomain.com* domain) *may* extract the domain part of the 'HELO' SMTP message as a secondary check;
- it queries DNS to receive the SPF record of the sending domain (in our case: *example.com*);
- it checks if the sender machine is in the authorized list mentioned in the SPF record;
- if the check is successful, the mail is delivered to the addressee;
- if the check fails, the action mentioned in the SPF record is taken (in general, the mail is rejected or marked as spam. See the next paragraph to have a list of the potential actions).

Setting up SPF for a domain

An SPF record is a DNS (Domain Name System) TXT record containing information about authorized mail servers. The SPF syntax allows you to specify the list of servers allowed or forbidden to send emails on behalf of your domain,

There can only be one SPF record per domain and it must be less than 512 characters long.

The complete syntax of the SPF record is complicated; you can refer to RFC 7208 for a complete definition.

The general syntax of an SPF record is the following:

```
<domain> TXT "v=spf1 <action><list of servers>..."
```

Example 1

Let's assume we create a SPF record for the *example.com* domain.

Let's assume the only legit mail server for this domain is *smtp.example.com*.
The SPF record will look like this:

```
example.com TXT "v=spf1 +a:smtp.example.com -all"
```

- ***v=spf1*** identifies the record as being an SPF record (there can be many other uses for a DNS TXT record).
- **+** means that, if the sending host matches with the following list, it is considered as authorized, and the mail must be delivered as is. Note that the + sign can be omitted.
- ***a:*** means that the name that follows is a DNS name for a server (or service).
- ***smtp.example.com*** is the DNS name of an allowed server.
- **-** means that, if the sending host matches with the following list, it is considered as **not** authorized, the spf status is 'failed', and the mail should be rejected, placed in quarantine or flagged as spam, depending on the policy put in place at the receiving end.
- ***all*** means all servers that did not satisfy the previous entries in the SPF record. It should be the latest entry in the record; all following entries are ignored.

If the mail server has no DNS name but an IP address (for example 109.88.52.201), the record should be modified as follows:

```
example.com TXT "v=spf1 +ip4:109.88.52.201 -all"
```

The ip4 mechanism allows a single IP address, or a subnet address in CIDR notation (e.g. 109.88.52.0/24 = all IP addresses between 109.88.52.0 and 108.88.52.255).

Example 2

Let's assume *serviceprovider.com* is an Internet Service Provider providing mail hosting, and has a client whose domain name is *example.com*.

Let's assume the system administrator for *example.com* wants to create an SPF record for his domain, only allowing one of its onsite servers and his provider email servers. To test his record, he also does not want that mails from unauthorized servers are rejected, but he wants them marked as dubious so he can analyze them.
The SPF record will look like this:

```
example.com TXT "v=spf1 +a:smtp.example.com
+mx:serviceprovider.com ~all"
```

Like in the previous example, he allows the *smtp.example.com* server, and allows the mail server of his service provider by telling SPF to trust the servers mentioned in the MX record of the serviceprovider.com domain.

The last part has a '~' action. This means that mails from other servers are marked as 'softfail'. This allows the receiver to mark mails as dubious, but not completely reject them.

Usually, a service provider has multiple servers, changing over time, with potentially complex rules. In this case, the best solution is just to include his SPF record:

```
example.com TXT "v=spf1 +ip4:109.88.52.201
include:spf.serviceprovider.com ~all"
```

If you use an Internet service provider or a domain registrar, they will document how you have to configure your SPF record to include their mailing infrastructure.

Finding the SPF record for a domain

As the SPF record is a DNS TXT record, you can get it with the following commands:

```
dig -t txt <domain>
nslookup -ty=txt <domain>
```

You will receive all the TXT records for the domain; look for one beginning with v=spf1. For example, let's find the SPF record for the FBI:

```
nslookup -ty=txt <domain>nslookup -ty=txt fbi.gov | grep spf
fbi.gov text = "v=spf1 +mx ip4:153.31.0.0/16 -all"
```

The answer is pretty straightforward. The allowed servers are:

- all servers listed by the MX records of the *fbi.gov* domain
- all servers having an IP address between 153.31.0.0 and 153.31.255.255

```
nslookup -ty=mx fbi.gov
Server: 9.9.9.9
Address: 9.9.9.9#53

Non-authoritative answer:
fbi.gov mail exchanger = 20 mx-west.fbi.gov.
fbi.gov mail exchanger = 10 mx-east.fbi.gov.


nslookup mx-west.fbi.gov
Server: 9.9.9.9
Address: 9.9.9.9#53


Non-authoritative answer:
Name: mx-west.fbi.gov
Address: 153.31.192.142


nslookup mx-east.fbi.gov
Server: 9.9.9.9
Address: 9.9.9.9#53
```

```
Non-authoritative answer:
Name: mx-east.fbi.gov

nslookup -ty=mx fbi.gov
Server: 9.9.9.9
Address: 9.9.9.9#53


Non-authoritative answer:
fbi.gov mail exchanger = 20 mx-west.fbi.gov.
fbi.gov mail exchanger = 10 mx-east.fbi.gov.


nslookup mx-west.fbi.gov
Server: 9.9.9.9
Address: 9.9.9.9#53


Non-authoritative answer:
Name: mx-west.fbi.gov
Address: 153.31.192.142


nslookup mx-east.fbi.gov
Server: 9.9.9.9
Address: 9.9.9.9#53


Non-authoritative answer:
Name: mx-east.fbi.gov
Address: 153.31.119.142
```

You can also use the following free websites to find SPF records:

- MXToolbox
- EasyDmarc
- Spf-record

Checking SPF results in email headers

The SPF protocol adds *Received-SPF:* headers in the emails it processes.

The general syntax is:

```
Received-SPF: <comment> <result> <explanation of result>
```

where *<result>* is pass, fail, softfail, neutral, none, temperror, permerror.

- ***temperror*** and ***permerror*** indicate that there were errors in querying or processing the DNS requests.
- ***none*** usually means that the sender domain does not have en SPF record

- ***neutral*** do not give any positive or negative indication.

Let's check a real world mail header. This is an answer from the *flyordie.com* site for a password reset request sent from *pltrash2@gmail.com*.

```
Delivered-To: pltrash2@gmail.com
Received: by 2002:a05:7412:b412:b0:fa:52c5:c6f5 with SMTP id
du18csp2235648rdb;
Sun, 11 Feb 2024 15:13:47 -0800 (PST)
X-Google-Smtp-Source:
AGHT+IGILZFYicIi2vRQIu4ua9M2m20+4BIDGMJFY3OO+4pT2Mv5POsOoU4M/cS+f0
IvCoIlCNJt
X-Received: by 2002:a0c:cd13:0:b0:68c:4774:a9ea with SMTP id b19-
20020a0ccd13000000b0068c4774a9eamr6013509qvm.46.1707693227575;
Sun, 11 Feb 2024 15:13:47 -0800 (PST)
ARC-Seal: i=1; a=rsa-sha256; t=1707693227; cv=none;
d=google.com; s=arc-20160816;
b=doZ/SOhBSyLRit5OySbs8hHay3NBWzZMYIeSczIBcHt2Y7/
NfawG3Oi9bHIf5EX2pm
qO+rHrfLcSmaPZYTnUfBM34qzoqDqVx2Osk9eiAX+CR1Ze8zZgJ7z0nsOGgHIAuL+G
AR
p+tHbWmP8s3tcPzXslogNw/VFiSYfD1escun/rkRStMOHrwSavyI2CWCGg+Y/j/
7I2/r
mH4F7UkQQf3Dbma3jvSsdTCFKniWQXg+cEwjJrdwlcg2GYeD6ox9WsSVW4br2243uU
ah
QSuu9nBrr8dn4hOqJKrfBvuvt75rySEud6OffjnGYRkyygJMdYvjD9zPDpR8E4NQqG
Ly
83vw==
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed;
d=google.com; s=arc-20160816;
h=content-transfer-encoding:mime-version:subject:message-
id:to:sender
:from:date;
bh=53dCbGE549Tr46FbHHEtKSI+UEYOOXdSSfVza3t/Jxo=;
fh=sP5OreUJhcIZXyhBOHrOOm5IvxQIjTB/JbpR2KvlL2O=;
b=z1OUPczwmJahzL4qloKQQob79peTUgZDeYrDBdEGA79FjGiPg4ONyXnN1EjO7LQc
qZ
HjJxiEEHz+d2iHwgfo3KhFZTmmFg9goBZpby/LK6LaaOkNxf58t+objp/
65mglPHgnKO
GJScfA+I41OqPRuER+RrwqbRIWXOzdQ2O2sUqFS28/TDrIm8/
g6c4EoQxgzMm31QbMEC
jvDoucWgSIreksOJkwMoIG9xZynjdQUe5K+Zpsbsx/
C5iD3SwgN8Ds5gsNEEns6myzMR
```

```
HfiyCYuJmjOipXKCi8Tqwm/JCMT49//g6DHCdoVXf+/
xB7TaibK4EYYbbDnnOu4eXIu4
WTwg==;
dara=google.com
ARC-Authentication-Results: i=1; mx.google.com;
spf=pass (google.com: domain of no-reply@flyordie.com designates
208.167.241.84 as permitted sender) smtp.mailfrom=no-
reply@flyordie.com
Return-Path: <no-reply@flyordie.com>
Received: from regmail2.flyordie.com (regmail2.flyordie.com.
[208.167.241.84])
by mx.google.com with SMTP id gm9-
20020a056214268900b0068cb633507asi7586096qvb.575.2024.02.11.15.13.
47
for <pltrash2@gmail.com>;
Sun, 11 Feb 2024 15:13:47 -0800 (PST)
Received-SPF: pass (google.com: domain of no-reply@flyordie.com
designates 208.167.241.84 as permitted sender) client-
ip=208.167.241.84;
Authentication-Results: mx.google.com;
spf=pass (google.com: domain of no-reply@flyordie.com designates
208.167.241.84 as permitted sender) smtp.mailfrom=no-
reply@flyordie.com
Received: from pipa1.in.flyordie.com (pipa1 [192.168.100.8])
by regmail2.flyordie.com
with SMTP (Mireka 4.0) id LSI57QZN
for pltrash2@gmail.com;
Mon, 12 Feb 2024 00:35:02 +0100 (CET)
Received: from pipa1 (localhost [127.0.0.1])
by pipa1.in.flyordie.com
with SMTP (Mireka 4.3.0) id LSI4GER2
for pltrash2@gmail.com;
Mon, 12 Feb 2024 00:13:46 +0100 (CET)
Date: Mon, 12 Feb 2024 00:13:46 +0100 (CET)
From: FlyOrDie <no-reply@flyordie.com>
Sender: FlyOrDie <no-reply@flyordie.com>
To: plecbe <pltrash2@gmail.com>
Message-ID: <1326803458.3842.1707693226668.JavaMail.tomcat@pipa1>
Subject: FlyOrDie password successfully reset
MIME-Version: 1.0
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: 7bit
```

The SPF header is:

```
Received-SPF: pass (google.com: domain of no-reply@flyordie.com
designates 208.167.241.84 as permitted sender) client-
ip=208.167.241.84;
```

The result is pass, and the explanation is that this mail was sent from the server with IP address 208.167.241.84.

Let's check the *flyordie.com* SPF record:

```
nslookup -ty=txt flyordie.com| grep spf
flyordie.com text = "v=spf1 ip4:82.192.93.216 ip4:82.192.93.217
ip4:82.192.93.218 ip4:208.167.241.84 -all"
```

208.167.241.84 is indeed in the list of authorized senders. I am now sure this mail was sent from a mail server authorized by flyordie.

Remember: SPF only checks the sender server is authorized by the sender domain owner, **NOT** the identity of the sender itself. The domain of the sender server can be different from the domain of the person who sends the email, which you see in the From: field of your email. It is only a **partial** validation.