

Ataque de E-mail Phishing + Spoofing (SPF, DKIM, DMARK)

ReadMe :

El siguiente documento dará comienzo con una breve descripción acerca del phishing y sus técnicas de ataque, para luego explicar a detalle el tema de "Spoofing + vulnerabilidades en protocolos de autenticación SPF, DKIM, DMARK".

[🔗] ¿En qué consiste el ataque del Phishing?

El medio para realizar este ataque es el correo electrónico.

Consiste en que un atacante envía un correo electrónico a su víctima para engañarla [1] y manipularla [2] a que entregue información privada o persuadir a que descargue y ejecute archivos o redirecciones (URLs) maliciosas.

[🔧] Técnicas empleadas en el ataque:

[1] Suplantación de identidad/spoofing:

El spoofing consiste en una técnica de suplantación de identidad, que en este caso se realiza a través de correo electrónico (email spoofing).

Se trata de utilizar la identidad de un usuario o entidad válida para obtener la confianza de la víctima y cometer delitos a posteriori.

[2] Ingeniería Social:

La ingeniería social es un ataque que intenta manipular a las personas para que realicen acciones a favor del atacante. La ingeniería social explota la naturaleza humana, aprovechando la disposición de las personas a ayudar o sus debilidades emocionales.

[✉] E-mail Spoofing

En este artículo nos centraremos en explicar la técnica de email spoofing.

Como mencionamos anteriormente el "spoofing" busca suplantar la identidad de un usuario o entidad. A continuación, se mencionan dos técnicas de suplantación.

>> Homóglifos (caracteres similares)

Los atacantes pueden utilizar caracteres de texto que son muy similares o incluso idénticos a los caracteres de texto legítimo. Por ejemplo, puede ser difícil distinguir entre una O ("O" mayúscula) y el 0 (número cero) o una l ("L" minúscula) y un 1 (número uno). Estos pueden utilizarse en los correos electrónicos de suplantación de identidad para hacerlos muy convincentes.

>> Vulnerabilidad en protocolos de autenticación (SPF, DKIM, DMARC)

El ataque de spoofing puede ocurrir cuando los atacantes aprovechan vulnerabilidades en la mala configuración del servidor de correo electrónico o en los protocolos de autenticación de correo electrónico, como **SPF** (Sender Policy Framework), **DKIM** (DomainKeys Identified Mail) y **DMARC** (Domain-based Message Authentication, Reporting & Conformance).

Estos protocolos están diseñados para prevenir la suplantación de identidad (spoofing) en correos electrónicos, pero si no se configuran correctamente, los atacantes pueden enviar correos falsificados que parecen provenir de una entidad legítima.

[] Tres protocolos de autenticación de remitente

>> SPF: Verifica la dirección IP del dominio de envío del correo.

>>DKIM: Verifica que el correo electrónico este firmado por el dominio de envío.

>> DMARC: Permite establecer políticas para el destinatario basadas en los resultados de SPF y DKIM y valida la dirección del correo electrónico en la cabecera from del correo.

[] SPF (Sender Policy Framework)

[Alice -> a.com -> alice@a.com | Bob -> b.com -> bob@b.com]

1) Alice le envía un correo electrónico a Bob.

2) El SPF (por el lado de Bob) hace una consulta del dominio (domain = a.com) de Alice para obtener las direcciones IP habilitadas para enviar correos en nombre del dominio.

3) Luego verifica si la IP del remitente (alice@a.com) coincide con la lista de IPs habilitadas.

4) Para que este proceso sea posible Alice tiene que publicar un archivo TXT en su DNS con las direcciones IPs habilitadas para enviar correos con el nombre de su dominio (a.com)

EJ:

a.com TXT "v=spf1 include:_spf.google.com include:spf.protection.outlook.com ip4:192.168.1.1 -all"

Desglose del valor:

1. **v=spf1**: Especifica que es un registro SPF versión 1.
2. **include:_spf.google.com**: Permite que los servidores de Google y Microsoft envíen correos en nombre de tu dominio y especifica la dirección ip habilitada. Cambia esta entrada según los servicios que uses (por ejemplo, _spf.example.com si usas otro proveedor).
3. **~all**: Indica que los correos no enviados por los servidores especificados deben ser marcados como *soft fail* (menos estricto). También puedes usar:
 - **-all**: Para rechazar estrictamente los correos no autorizados.
 - **+all**: Para permitir cualquier servidor (no recomendado por seguridad).

DKIM (DomainKeys Identified Mail)

[Alice -> a.com -> alice@a.com | Bob -> b.com -> bob@b.com]

1) (Por el lado de Alice) quien es la que envía el correo, publica una llave pública en su DNS.

2) Luego genera una firma DKIM con su llave privada y la adjunta en las cabeceras del correo enviado a Bob.

3) (Por el lado de Bob) se hace una consulta al dominio de (a.com) para obtener la llave pública de Alice y validar la firma recibida en el correo.

De esta forma se corrobora que el correo esta firmado por el dominio de envío habilitado.

EJ:

a.com TXT "v=DKIM1; k=rsa;
p=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAs1i4OKOvBYF...RRo+FQIDAQAB"

Desglose del registro:

1. **default._domainkey:**

- default: Es el *selector*, una etiqueta que permite gestionar múltiples claves DKIM para el mismo dominio. Puedes cambiarlo por otro nombre como mail o selector1.
- _domainkey: Es obligatorio y especifica que el registro pertenece a DKIM.

2. **v=DKIM1:** Versión de DKIM (siempre será DKIM1).

3. **k=rsa:** Tipo de criptografía utilizada (normalmente RSA).

4. **p=MIIBIjANBg...:** Es la clave pública utilizada para validar la firma del correo. Esta clave la genera tu proveedor de correo o tu propio servidor de correo.

[●] **DMARC (Domain-based Message Authentication, Reporting & Conformance)**

[Alice -> a.com -> alice@a.com | Bob -> b.com -> bob@b.com]

DMARC realiza un test de validación para identificar si el dominio que aparece en la cabecera del correo from (remitente = alice@a.com) coincide con el dominio que **SPF** y **DKIM** han validado.



Un correo electrónico pasa la validación DMARC cuando SPF y DKIM muestran resultados positivos y el dominio que figura en la cabecera from (alice@a.com) también fue validado por DMARC.

Cuando un usuario implementa el protocolo DMARC debe publicar un registro TXT en su DNS. Aquí se establece una política sobre que acción se toma con los correos que infrinjan los protocolos de autenticación (SPF, DKIM, DMARC).

Las acciones son las siguientes:

-> Los correos son rechazados (reject)

-> Los correos son entregados como spam (quarantine)

-> Los correos son entregados a la bandeja de entrada (none)

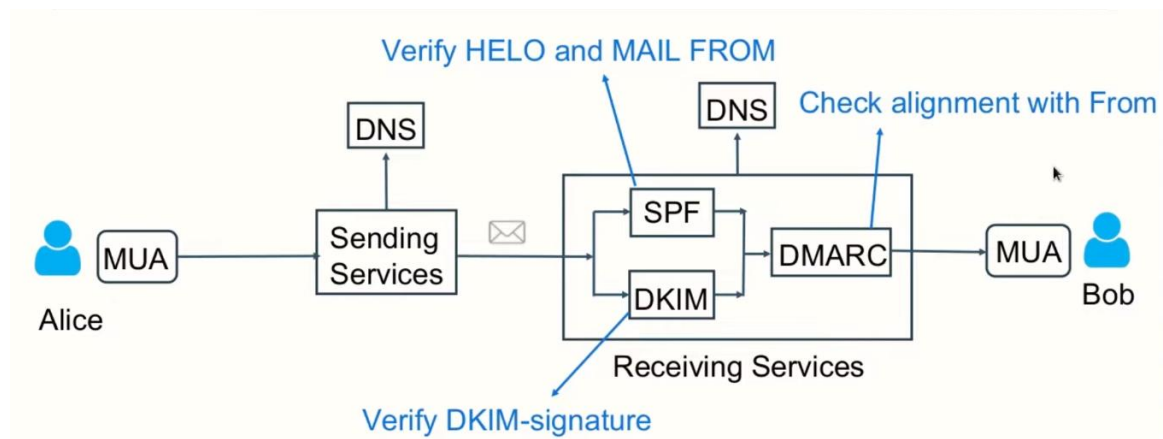
EJ:

b.com TXT "v=DMARC1; p=none; rua=mailto:reportes@tudominio.com"

Desglose del registro:

1. **v=DMARC1**: Especifica que el registro es un registro DMARC (obligatorio).
2. **p=quarantine**: Política de manejo para correos no autenticados:
 - **p=none**: No hace nada, solo monitorea (útil para pruebas iniciales).
 - **p=quarantine**: Envía los correos no autenticados a la carpeta de spam.
 - **p=reject**: Rechaza directamente los correos no autenticados (estricto, recomendado cuando todo está configurado correctamente).
3. **rua=mailto:reportes@tudominio.com**: Dirección de correo donde se envían los reportes agregados (estadísticas de autenticación).
4. **ruf=mailto:fallos@tudominio.com**: Dirección de correo donde se envían reportes detallados sobre fallos individuales (opcional).
5. **fo=1**: Opciones de generación de reportes de fallos:
 - **0**: Solo reportar fallos si tanto SPF como DKIM fallan (por defecto).
 - **1**: Reportar fallos si cualquiera de ellos (SPF o DKIM) falla.
 - **d**: Reportar si DKIM falla.
 - **s**: Reportar si SPF falla.

Procesos de autenticación de un correo electrónico



[🦴] Bypassing the Authentication

Vulnerabilidad en protocolos SPF y DMARC

1. Falta de un registro SPF

Si un dominio no tiene configurado un registro SPF, los servidores receptores no tienen forma de verificar si un servidor de correo tiene permiso para enviar correos en nombre de ese dominio. Esto permite a los atacantes:

- Usar cualquier servidor SMTP para enviar correos falsificados con direcciones MAIL FROM de ese dominio.
- Los correos parecen legítimos, ya que no hay verificación contra un registro SPF.

El atacante coloca su servidor (atacante) como autorizado para poder enviar un correo en nombre del dominio autorizado (a.com) y este le llega a la víctima con suplantación de identidad.

Hello: atacante.com

Mail From: atacante.com

From: alice@a.com

To: bob@b.com

Esto ocurre cuando directamente no hay un registro SPF activo.

2. Registro SPF mal configurado

Si un registro SPF está configurado incorrectamente, puede permitir a servidores no autorizados enviar correos en nombre del dominio. Ejemplos:

- **Inclusión excesiva de servidores:** Si el registro SPF incluye demasiados servidores (como un rango IP muy amplio con ip4:0.0.0.0/0 o incluye innecesarios), cualquier servidor en ese rango podría enviar correos falsificados.
- **Uso de mecanismos abiertos:**
 - Utilizar +all en un registro SPF permite que cualquier servidor pase la validación SPF, lo que hace que el registro sea prácticamente inútil.
 - Ejemplo de registro inseguro:
 - Copiar código
 - v=spf1 +all

RESULTADO: SPF: **pass**, DMARK: **pass**

3. Uso de subdominios sin SPF

Algunos atacantes apuntan a subdominios no protegidos con registros SPF. Si el dominio principal tiene un registro SPF pero un subdominio no lo tiene, los atacantes pueden usar ese subdominio para falsificar correos.

4. Validación débil en el servidor receptor

Aunque un dominio tenga configurado SPF (a.com), si el servidor receptor (b.com):


- **No realiza la validación SPF**, ignorará los resultados y aceptará correos falsificados.
- **No actúa ante fallos de SPF**, como aceptar correos incluso si el resultado de la verificación es fail.
- **No tiene o tiene mal configurado DKIM y DMARK**, ignora los resultados y permite ingresar los correos (malignos) a la bandeja de entrada de la víctima.

RESULTADO: SPF: **pass**, DMARK: **pass**

Ejemplo de un ataque de spoofing aprovechando SPF:

1. El dominio `example.com` tiene un registro SPF inseguro:


css

 Copiar código

```
v=spf1 +all
```

2. Un atacante utiliza cualquier servidor SMTP para enviar un correo con:

sql

 Copiar código


```
MAIL FROM: <admin@example.com>
```

3. El correo pasa la validación SPF debido al `+all`, y el servidor receptor lo entrega como si fuera legítimo.
4. La víctima recibe un correo que parece auténtico pero proviene del atacante.

Buenas prácticas para evitar spoofing con SPF:

1. Configura un registro SPF restrictivo, autorizando solo servidores específicos:

makefile


 Copiar código

```
v=spf1 ip4:192.0.2.1 include:spf.example.net -all
```

- Usa `-all` (fail) en lugar de `~all` (soft fail) para denegar servidores no autorizados.

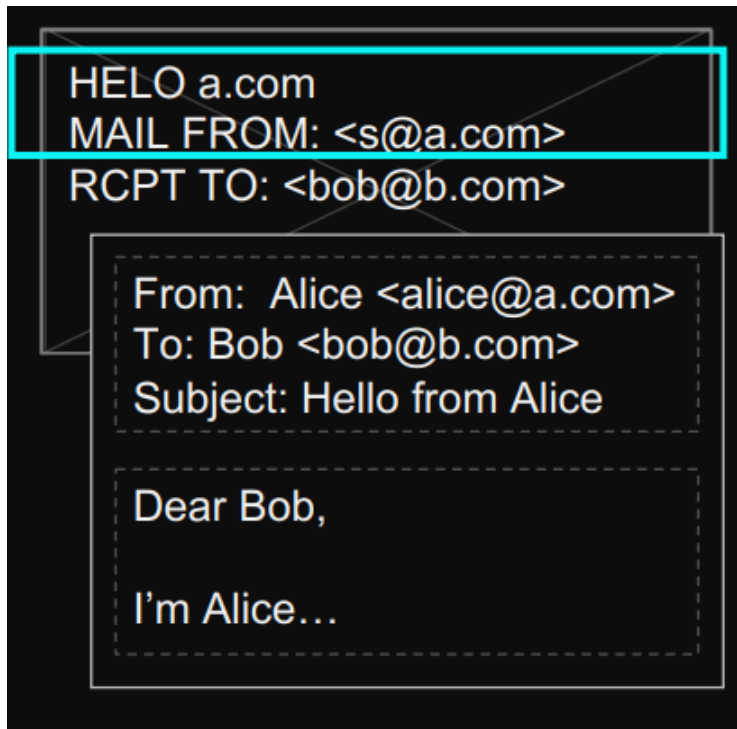
2. Implementa **DKIM** para firmar correos electrónicos y asegurar su integridad.
3. Configura **DMARC** para definir políticas claras de manejo de correos no autenticados:

css

 Copiar código

```
v=DMARC1; p=reject; rua=mailto:dmARC-reports@example.com
```

4. Monitorea regularmente registros SPF, DKIM y DMARC para garantizar que estén actualizados.



[💀] Vulnerabilidad en DKIM y DNS

Esto ocurre cuando el atacante firma el correo con su clave pública (servidor atacante). Y el receptor (b.com) cuando recibe el correo, valida la firma del servidor atacante, ya que el carácter **\x00 (null byte)** genera una mala interpretación por parte del servidor DNS afectado.

2. La vulnerabilidad con \x00 en DKIM y DNS

En este ataque, un atacante explota una debilidad en cómo algunos servidores y resolutores DNS interpretan las consultas relacionadas con subdominios, combinada con una mala validación de firmas DKIM. Aquí está el proceso típico del ataque:

Paso 1: Creación de un registro malicioso

1. El atacante controla un dominio malicioso, por ejemplo, attacker.com.
2. Configura un registro DKIM en el DNS de su dominio, que incluye su propia clave pública.
 - Por ejemplo, en default._domainkey.attacker.com.

Paso 2: Modificación del encabezado DKIM del correo

El atacante envía un correo electrónico falso desde un dominio legítimo, pero modifica el encabezado DKIM-Signature para que apunte a un subdominio malicioso con un terminador null \x00. Ejemplo:

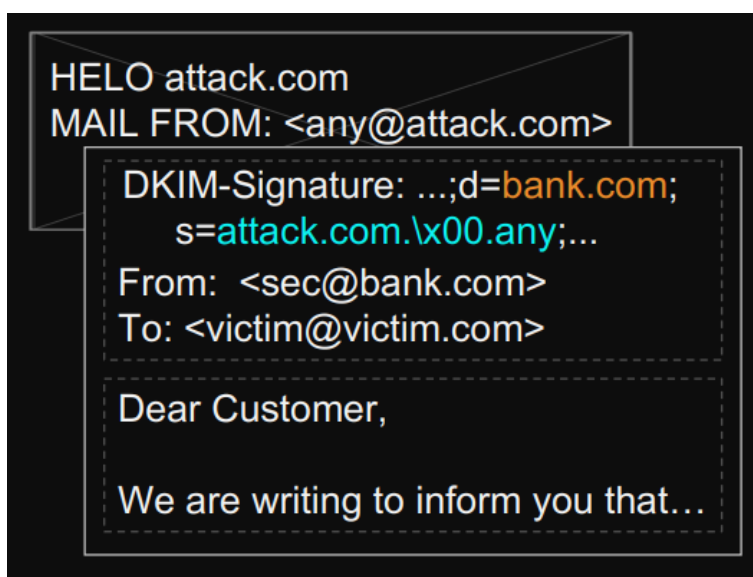
```
arduino Copiar código  
d=legitdomain.com\x00.attacker.com; s=default;
```

- Aquí, d=legitdomain.com\x00.attacker.com es clave:
 - Algunos resolutores DNS interpretan el null byte \x00 como un terminador.
 - Esto hace que la consulta DNS **finalice en legitdomain.com para algunos sistemas** y en **attacker.com para otros**.

Paso 3: Explotación de la vulnerabilidad

1. El servidor receptor realiza una consulta DNS para obtener la clave pública del dominio que aparece en d=.
2. Debido al uso del \x00:
 - Algunos resolutores buscan en el dominio malicioso attacker.com, donde encuentran la clave pública del atacante.
 - El receptor usa esa clave para validar la firma DKIM, lo que lleva a una validación exitosa.
3. El correo falsificado pasa las verificaciones de DKIM, engañando al receptor.

RESULTADO: DKIM pass, DMARC pass



[👹] **Vulnerabilidad Duplicado de Cabezeras**

En este ataque el agresor duplica la cabecera “from” del correo electrónico para forzar a que el servidor victima autentique el dominio emisor (attacker.com). De esta manera el protocolo SFP y DMARK validarían la autenticación del dominio del atacante.

3. ¿Cómo se usa este ataque?

Un atacante puede aprovechar estas inconsistencias para engañar al receptor o al sistema de correo. Aquí hay dos escenarios comunes:

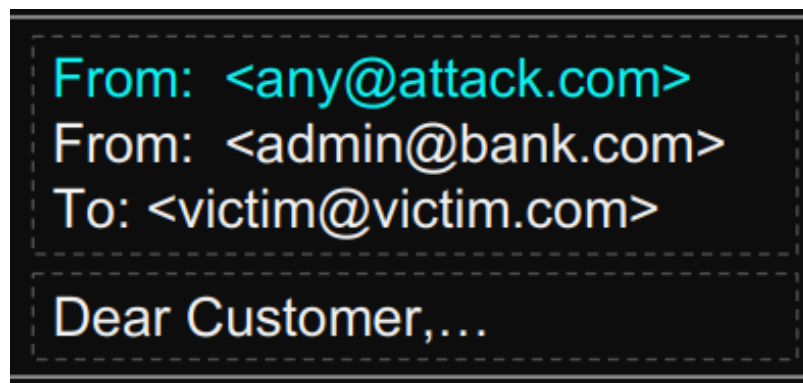
a) Engaño al cliente de correo (spoofing visual):

- Un cliente de correo puede mostrar el segundo encabezado From (que parece legítimo) mientras que el servidor utiliza el primero para la validación.
- Ejemplo:

From: attacker@malicious.com

From: victim@legitimate.com

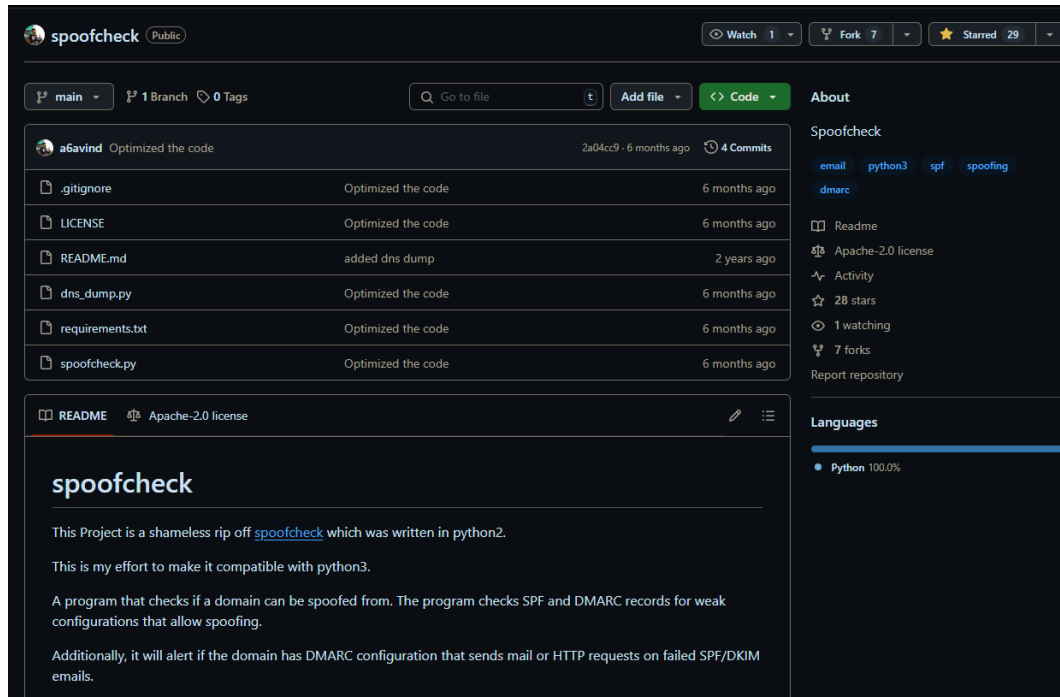
- El cliente muestra: victim@legitimate.com (aparentemente legítimo).
- Los filtros de seguridad no detectan nada porque el sistema considera attacker@malicious.com como el verdadero remitente.



[🔧] Herramientas para auditar SPF y DMARC

>> Spoofcheck Tool

[Github] <https://github.com/a6avind/spoofcheck>



Un script de python3 que verifica si se puede suplantar un dominio. El programa verifica los registros SPF y DMARC en busca de configuraciones débiles que permitan la suplantación.

>> Aplicación

A continuación, se muestran capturas de pruebas realizadas a algunos dominios.

[Disclaimer]: La muestra del siguiente contenido tiene exclusivamente fines educativos. El objetivo es demostrar de manera práctica los conocimientos teóricos explicados anteriormente.

1# Ejemplo

Dominio: plataforma10.com

```
(venv)-[~]/tools/spoofcheck
$ python3 spoofcheck.py plataforma10.com
[*] Found SPF record:
[*] v=spf1 a:rivadavia-iplan.plataforma10.net a:rivadavia-fibertel1.plataforma10.net a:rivadavia-fibertel2.plataforma10.net include:_spf.google.com ip4:54.208.111.148 ~all
[*] SPF record contains an All item: ~all
[*] Found DMARC record:
[*] v=DMARC1; p=none; rua=mailto:administrator@plataforma10.com
[*] DMARC policy set to none
[*] Spoofing possible for plataforma10.com!
```

```

(venv)-( )-[~/tools/spoofcheck]
$ python3 spoofcheck.py plataforma10.com
[*] Found SPF record:
[*] v=spf1 a:rivadavia-iplan.plataforma10.net a:rivadavia-fiberte
[*] SPF record contains an All item: ~all
[*] Found DMARC record:
[*] v=DMARC1; p=none; rua=mailto:administrator@plataforma10.com
[+] DMARC policy set to none
[+] Spoofing possible for plataforma10.com!

```

Resultado: mala configuración del DMARC (p=none)

En este caso podríamos enviar un correo electrónico suplantando alguna identidad y el servidor de correo entregaría el mismo en la bandeja de entrada sin restricción.

2# Ejemplo

Dominio: aa2000.com.ar

```

(venv)-( )-[~/tools/spoofcheck]
$ python3 spoofcheck.py aa2000.com.ar
[*] Found SPF record:
[*] v=spf1 mx ip4:186.153.179.171 ip4:200.32.111.226 ip4:200.32.111.245 ip4:200.70.19.5 ip4:200.45.16.198 ip4:190.139.108.116 include:spf.protection.outlook.com exists:%{i}.spf.hc6303-47.ip
hmx.com include:spf.us.exclaimer.net -all
[*] SPF record contains an All item: -all
[*] Found DMARC records:
[*] v=DMARC1; p=quarantine; rua=mailto:ruadmarc@aeropuertosargentina.com
[-] DMARC policy set to quarantine
[*] Aggregate reports will be sent: mailto:ruadmarc@aeropuertosargentina.com
[-] Spoofing not possible for aa2000.com.ar

```

```

(venv)-( )-[~/tools/spoofcheck]
$ python3 spoofcheck.py aa2000.com.ar
[*] Found SPF record:
[*] v=spf1 mx ip4:186.153.179.171 ip4:200.32.111.226 ip4:200.32.111.245 ip4:20
hmx.com include:spf.us.exclaimer.net -all
[*] SPF record contains an All item: -all
[*] Found DMARC record:
[*] v=DMARC1; p=quarantine; rua=mailto:ruadmarc@aeropuertosargentina.com
[-] DMARC policy set to quarantine
[*] Aggregate reports will be sent: mailto:ruadmarc@aeropuertosargentina.com
[-] Spoofing not possible for aa2000.com.ar

```

Resultado: buena configuración del DMARC (p=quarantine)

Para este caso si quisiéramos suplantar la identidad, el servidor de correo entregaría el mismo en la bandeja de spam.

3# Ejemplo

Dominio: mercadolibre.com.ar

```
(venv)-( )-[~/tools/spoofcheck]
$ python3 spoofcheck.py mercadolibre.com.ar
[*] Found SPF record:
[*] v=spf1 exists:%{i}._spf.mta.salesforce.com include:spf.mercadolibre.com.ar ip4:172.217.128.0/19 ip4:172.217.160.0/20 ip4:172.217.192.0/19 ip4:18.211.176.124 ip4:18.211.61.67 ip4:18.213.72.148 ip4:184.73.44.157 ip4:35.190.247.0/24 ip4:64.233.160.0/19 ~all
[*] SPF record contains an All item: ~all
[*] Found DMARC record:
[*] v=DMARC1; p=reject; rua=mailto:dmarcrua@mercadolibre.com, mailto:mercadolibre@rua.netcraft.com, mailto:dmarc_agg@vali.email; ruf=mailto:dmarcruf@mercadolibre.com, mailto:mercadolibre@ruf.netcraft.com; adkim=s; aspf=r; rf=afrf; pct=100;
[-] DMARC policy set to reject
[*] Aggregate reports will be sent: mailto:dmarcrua@mercadolibre.com, mailto:mercadolibre@rua.netcraft.com, mailto:dmarc_agg@vali.email
[*] Forensics reports will be sent: mailto:dmarcruf@mercadolibre.com, mailto:mercadolibre@ruf.netcraft.com
[-] Spoofing not possible for mercadolibre.com.ar
```

```
(venv)-( )-[~/tools/spoofcheck]
$ python3 spoofcheck.py mercadolibre.com.ar
[*] Found SPF record:
[*] v=spf1 exists:%{i}._spf.mta.salesforce.com include:spf.mercadolibre.com.ar ip4:172.217.128.0/19 ip4:172.217.160.0/20 ip4:172.217.192.0/19 ip4:18.211.176.124 ip4:18.211.61.67 ip4:18.213.72.148 ip4:184.73.44.157 ip4:35.190.247.0/24 ip4:64.233.160.0/19 ~all
[*] SPF record contains an All item: ~all
[*] Found DMARC record:
[*] v=DMARC1; p=reject; rua=mailto:dmarcrua@mercadolibre.com, mailto:mercadolibre@rua.netcraft.com; adkim=s; aspf=r; rf=afrf; pct=100;
[-] DMARC policy set to reject
[*] Aggregate reports will be sent: mailto:dmarcrua@mercadolibre.com, mailto:mercadolibre@rua.netcraft.com, mailto:dmarc_agg@vali.email
[*] Forensics reports will be sent: mailto:dmarcruf@mercadolibre.com, mailto:mercadolibre@ruf.netcraft.com
[-] Spoofing not possible for mercadolibre.com.ar
```

Resultado: buena configuración del DMARC (p=reject)

Para este caso si quisiéramos suplantar la identidad, el servidor de correo rechazaría el envío del correo suplantado.

>> Prueba con EMKEI

Por medio de esta herramienta web (emkei.cz) podremos realizar pruebas para poner a prueba la configuración de los servidores web víctimas.

Emkei es una herramienta online y de uso libre, sin embargo, la dirección IP del servidor de Emkei no está habilitada para enviar correos en nombre de nuestro dominio y además su dirección IP puede estar en la blacklist de algunos servicios de correo. Por lo tanto, los correos se entregarán siempre como “spam” debido a los protocolos de autenticación (SPF y DKIM).

A pesar de esto, los correos se envían y se entregan sin ningún problema. Pero siempre obtendremos una advertencia de spam por parte de los servidores de correo destinatarios, al menos que este mal configurado.

Una solución podría ser montar un servidor en un hosting en el cual carguemos un archivo que ejecute el envío de un correo de spoofing.

#Paso 1

Enviar el spoofing email con Emkei

FAKEE'S MAILER

Free online fake mailer with attachments, encryption, HTML editor and advanced settings...

✔ E-mail sent successfully

From Name: Aeropuerto Argentina 2000

From E-mail: rrhh@aa2000.com.ar

To: [redacted]

Subject: Calificación a entrevista de empleo

Attachment: Browse... No file selected.
Attach another file
Advanced Settings

Content-Type: ☒ text/plain ☐ text/html ☐ Editor

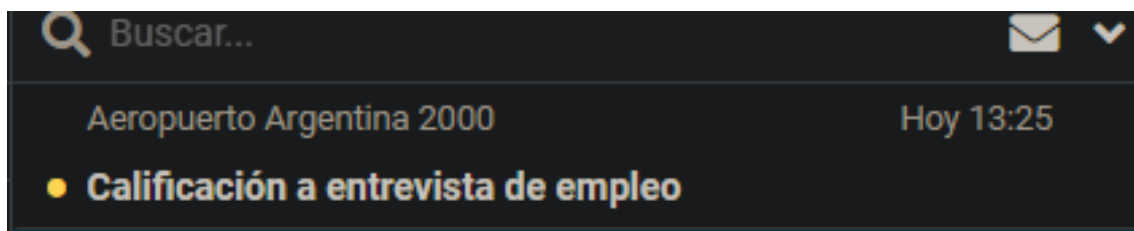
Text: Buenos días, usted a sido calificado para ser entrevistado por nuestro equipo de recursos humanos.

Captcha: ☐ I am human  hCaptcha Privacy - Terms

Send **Clear**

#Paso 2

Recibimos el spoofing email en nuestra casilla víctima.



Descripción del mensaje.

Calificación a entrevista de empleo



De Aeropuerto Argentina 2000 <rrhh@aa2000.com.ar>

Destinatario [redacted]

Responder a rrhh@aa2000.com.ar

Fecha Hoy 13:25

Prioridad Normal

Resumen Cabeceras

Buenos días, usted a sido calificado para ser entrevistado por nuestro equipo de recursos humanos.

#Paso 3

Verificamos los protocolos de autenticación en las cabeceras del correo.

```
dkim=none;  
dmarc=fail reason="No valid SPF; No valid DKIM" header.from=aa2000.com.ar (policy=quarantine);  
spf=fail ([redacted]: domain of rrhh@aa2000.com.ar does not designate 114.29.236.247 as permitted sender) smtp.mailfrom=rrhh@aa2000.com.ar
```

```
dkim=none;  
dmarc=fail reason="No valid SPF; No valid  
spf=fail ([redacted]: domain
```

#Paso 4

El protocolo SPF=fail, es debido a que la dirección IP de Emkei no esta habilitada por el dominio de aa2000.

PTR [SHOW RAW]

Type	Domain Name	TTL	Canonical Name
PTR	114.29.236.247	14400	emkei.cz.

Conclusión:

En este caso el email spoofing que ejecutamos se entregó en la casilla de correo de spam, ya que nuestro servidor de destino está bien configurado con la política DMARK=quarantine.

Sin embargo, si el servidor de destino estuviera mal configurado y con el DMARK=none, nuestro mensaje de spoofing podría haber tenido éxito y ser entregado en la bandeja de entrada. Aumentando más las chances de infectar a la víctima.

FUENTES:

https://www.youtube.com/watch?v=oKkTM5fJWWg&list=PLSgzl4cWqQ6qR86geAOdgO5dpy_9bRAcG&index=37

https://www.usenix.org/system/files/sec20_slides_chen-jianjun.pdf

Contacto

 <https://www.linkedin.com/in/david-padron-9a74aa323/>

 <https://github.com/FeathersMcgr4w>

 <https://feathersmcgr4w.github.io/cyber-portfolio/>