Anatomía de un ciberataque por SMTP

Miles de rebotes inundaron un buzón en minutos. No era suplantación: alguien había averiguado las contraseñas y enviaba desde el propio servidor.

Empezó una tarde de septiembre, con un aviso humano y no con una alerta automática.

Una persona de la empresa cliente escribió a Andy diciendo que el correo se había vuelto loco. Borraba los mensajes y volvían a llegar. Sus correos antiguos ya no aparecían. No paraba.

Adjuntó capturas. Todas mostraban lo mismo, repetido hasta llenar la pantalla: Undelivered Mail Returned to Sender.

La hipótesis equivocada

Ante una avalancha de rebotes, el manual dice una cosa: *backscatter*.

Alguien envía spam desde fuera poniendo tu dirección como remitente falso. Los destinatarios lo rechazan. Y como creen estar avisando al culpable, te mandan a ti el aviso de error. Miles de veces.

Encajaba con los síntomas, pero también encajaba la posibilidad contraria: que la cuenta estuviera comprometida y el spam saliera del propio servidor. Desde el buzón de la víctima, ambos escenarios se ven igual.

Cambiar la contraseña es lo que los separa. Si el envío se detiene, la cuenta estaba comprometida. Si continúa, la suplantación viene de fuera. Andy la cambió.

El envío desde esa cuenta se detuvo. Pero los rebotes seguían llegando.

La cola de salida del servidor explicó por qué:

```
mailq | tail -n 2
```

La respuesta fue `44528 Kbytes in 34353 Requests`.

Treinta y cuatro mil mensajes esperando salir. Los correos no venían de fuera. Salían de casa.

Por qué rebotaban

Este es el punto que casi siempre se cuenta mal.

Los mensajes no rebotaban porque nadie detectara una falsificación. No había ninguna que detectar. Salían autenticados desde la IP legítima, firmados correctamente, con todos los papeles en regla.

Rebotaban por motivos mucho más terrenales, escritos uno a uno en la propia cola:

- **Demasiado volumen.** Gmail contestaba `421-4.7.28` y Yahoo `421 4.7.0 TSS04`. Traducido: esto huele a spam, frena.
- **Mala reputación.** T-Online respondía `554 None/bad reputation`, negándose directamente a conversar con el servidor.

- **Destinatarios fantasma.** Buena parte de la lista apuntaba a dominios cuyo servidor de correo lleva años sin existir.

Es decir: el sistema inmunitario del correo mundial reaccionó exactamente como debía. Cada rechazo generó un aviso de vuelta hacia el remitente.

El remitente era real. Por eso los avisos aterrizaron en un buzón real.

El archivo de 256 megabytes

Para saber qué cuentas estaban comprometidas solo hay un sitio donde mirar, y Andy fue directo a él: el registro de autenticaciones del servidor de correo.

El primer intento no encontró nada. Y ahí apareció una trampa que merece la pena conocer: los registros rotan. El archivo activo solo contenía lo ocurrido desde medianoche, y el ataque había sido la tarde anterior.

```
ls -lt /var/log/maillog*
```

El listado mostró un archivo de 1 MB junto a otro de 256 MB. La diferencia de tamaño ya era, por sí sola, el relato de lo ocurrido.

```
grep -o 'sasl_username=[^ ,]*' /var/log/maillog.processed | sort | uniq
-c | sort -nr | head
```

El recuento no dejaba lugar a interpretaciones. Cinco buzones acumulaban entre mil cien y tres mil ochocientas autenticaciones cada uno. Más de diez mil envíos en total, frente a las pocas decenas que registraban esas mismas cuentas en su uso habitual.

No hubo suplantación. Alguien tenía las contraseñas.

Un intruso que cambia de puerta

Lo más revelador del incidente vino después de cada contención.

El corte de la primera cuenta duró poco. Minutos más tarde la cola volvía a llenarse, pero con otro remitente: una cuenta distinta, de otro dominio del mismo servidor.

Se cambió también esa contraseña. Apareció una tercera.

El detalle que lo explica todo es que cada cuenta enviaba a una lista diferente. Una apuntaba a direcciones institucionales de Ecuador. Otra, a un listado de correos alemanes de proveedores desaparecidos hace años. Eran campañas separadas lanzadas desde el mismo juego de llaves.

Quien estaba dentro no había acertado una contraseña: tenía varias, de dominios distintos, y las fue gastando a medida que se le cerraban las puertas.

Con la última cerrada, tocaba limpiar.

```
postsuper -d ALL
systemctl restart postfix
```

Se borraron 34.304 mensajes. El reinicio no es opcional: cambiar una contraseña no cierra las sesiones que ya están abiertas.

Después, el buzón que había recibido el aluvión. Veintiún mil seiscientos setenta y siete rebotes, imposibles de borrar a mano:

```
doveadm expunge -u buzon@dominio.tld mailbox INBOX HEADER Subject
"Undelivered Mail Returned to Sender"
```

Se hacía de noche. El servidor había dejado de enviar, la cola estaba vacía y el buzón limpio. La urgencia había terminado, pero el trabajo no.

La parte que no aparece en ningún registro

Hubo un momento del incidente que ningún log recoge.

El correo estaba comprometido, pero la conversación con la empresa iba por WhatsApp, así que el canal era fiable y la identidad de quien pedía la nueva clave no estaba realmente en duda.

Aun así, antes de enviarla, Andy preguntó cómo se llamaba la gata de su interlocutora.

No tenía gata. Tenía perra, y lo dijo. Esa era la respuesta

correcta.

Lo habitual en una verificación de identidad es preguntar algo que ambas partes conocen. Aquí la pregunta iba con trampa, más por gusto que por necesidad, pero el ejemplo enseñó en treinta segundos lo que una explicación técnica no habría transmitido igual de bien: que las credenciales se comprueban, y que una respuesta correcta puede estar preparada de antemano.

Las dos personas al otro lado lo captaron a la primera.

Lo que estaba bien y no bastó

El servidor tenía SPF con `-all`, DKIM firmando y DMARC publicado en todos los dominios. Es una base correcta.

Y protege exactamente de lo que fue la primera hipótesis: que alguien de fuera se haga pasar por el dominio.

Frente a este ataque no hizo nada. No podía hacerlo.

SPF, DKIM y DMARC protegen la identidad del dominio frente al exterior. Ninguno de los tres impide que alguien entre con la contraseña correcta.

Endurecerlo no resolvía nada de lo ocurrido, pero esa misma noche la política DMARC pasó a `p=reject` en todos los dominios. De paso apareció un registro SPF que aún apuntaba al nombre anterior del servidor. No estaba fallando, porque un wildcard en las DNS lo seguía resolviendo bien, y no tuvo nada que ver con el incidente. Se actualizó igualmente: es mejor tenerlo bien que tenerlo casi bien.

Revisar el DMARC de todos los dominios de golpe lleva un segundo:

```
for d in $(plesk db -Ne "SELECT name FROM domains WHERE status=0 AND parentDomainId=0"); do printf "%-25s " "$d:"; dig +short TXT _dmarc.$d; done
```

Conviene una matización, porque es una decisión consciente y no un descuido: esa política estricta gobierna el correo **saliente**. En la recepción, el servidor mantiene deliberadamente una comprobación permisiva.

El motivo es simple. No se puede controlar cómo tienen configurado

el correo los clientes y proveedores. Una regla rígida a la entrada acabaría tirando mensajes legítimos de terceros con la casa mal montada, y perder un correo de un cliente es un problema mayor que colar uno dudoso.

Las dos puertas sin vigilar

A la mañana siguiente, con el servidor en calma y sin nada ardiendo, tocaba la parte que no se puede hacer con prisa: averiguar por qué había sido posible.

Aparecieron dos carencias. Son el motivo de que esto llegara a treinta y cuatro mil correos.

La primera: nadie vigilaba el correo. Fail2ban llevaba años funcionando en el servidor, pero sus reglas cubrían únicamente la web.

```
fail2ban-client status
```

Tres reglas activas. Ninguna relacionada con el correo. Un bot podía probar contraseñas contra el puerto SMTP de forma indefinida sin que nada lo interrumpiera.

Activar la vigilancia nativa de Plesk es una línea:

```
plesk bin ip_ban --enable-jail plesk-postfix
fail2ban-client status plesk-postfix
```

El bloqueo por defecto era de 600 segundos, que frente a un ataque automatizado es casi una cortesía: diez minutos después la misma IP vuelve a la carga. Se subió a una hora tras cinco intentos fallidos.

La segunda: no había techo de envío. Una cuenta comprometida podía emitir sin límite, que es precisamente lo que ocurrió.

En Plesk se configura desde *Herramientas y configuración › Control de correo saliente*, contando destinatarios en lugar de mensajes para que un envío masivo en copia oculta no se cuele por el hueco.

Los límites quedaron en 100 correos por buzón y hora, 500 por dominio y 1.000 por suscripción.

Con esa barrera puesta, el ataque se habría frenado solo al alcanzar el tope, y habría generado una notificación en lugar de treinta y cuatro mil mensajes en cola.

Contraseñas de otra época

Queda la pregunta incómoda: ¿cómo consiguió alguien varias contraseñas de varios dominios distintos?

La respuesta resultó ser la menos sofisticada. Esas claves se habían creado alrededor de una década atrás, cuando «contraseña segura» significaba ocho caracteres, una mayúscula y un número.

Una década es mucho tiempo. Tiempo para que la potencia de cálculo se multiplique, para que las herramientas de fuerza bruta se afinen, y sobre todo para que esa misma clave, o una variación evidente, aparezca en alguna de las filtraciones masivas que se acumulan y se venden.

No hizo falta ninguna genialidad. Solo esperar lo suficiente.

Si alguien lee esto y arrastra contraseñas de hace una década, ese es el aviso. No hace falta un incidente para justificar el cambio: la antigüedad basta. Lo que era razonable en 2016 hoy no lo es, del mismo modo que lo que hoy parece robusto no lo será en 2036.

Y la presión no va a aflojar. La computación cuántica obligará a rehacer los cimientos criptográficos de internet, aunque eso afecte a las firmas y a los certificados más que a la clave de un buzón. Para las contraseñas, la amenaza cercana es menos exótica y más eficaz: hardware barato, modelos capaces de generar candidatas plausibles a partir de lo que cualquiera publica sobre sí mismo, y un archivo de filtraciones que solo crece.

Las claves de este servidor se sustituyeron por otras mucho más largas y sin relación con nada adivinable. Tocará volver a mirarlas sin esperar otros diez años ni otro incidente.

Cierre

La comprobación final fue la más simple de todas:

```
postqueue -p
```

Mail queue is empty. Nada saliendo, nada pendiente, nada rebotando.

Ninguno de los tres fallos era sofisticado: un servicio de vigilancia que no cubría el correo, un envío sin techo y unas claves que llevaban una década sin tocarse. Ninguno lo detectó el servidor. Lo detectó una persona diciendo que su bandeja de entrada no paraba de pitar.

Un incidente se cierra cuando la cola queda vacía. El trabajo de verdad empieza después, revisando en frío qué lo hizo posible y corrigiéndolo antes de que vuelva a pasar. Esa segunda parte no la exige nadie y es la que marca la diferencia.

El perímetro estaba bien construido. Sólido, revisado, correcto. Solo tenía un defecto: miraba en la dirección equivocada.

Los comandos de este artículo son los que se ejecutaron. Cualquiera con acceso a su propio servidor puede repetirlos para comprobar si tiene las mismas puertas sin vigilar.
