Alerta ID:	094
Fecha del reporte:	24/04/2026
Entidad:	Todas las entidades del ecosistema digital
Título:	Fuzzing como técnica de evasión en spam y phishing
Herramienta de detección	N/A
Activo involucrado:	Correo electrónico
Tipo de alerta:	Alerta
Nivel de riesgo:	Alto

Objetivo:

Informar a las organizaciones del ecosistema digital, sobre el uso cada vez más frecuente de técnicas de fuzzing en campañas de spam y phishing, empleadas para modificar de forma controlada los elementos clave del correo electrónico — como el asunto, el nombre del remitente y fragmentos del cuerpo del mensaje— con el fin de evadir los mecanismos tradicionales de detección basados en firmas y reputación, así como destacar su impacto en la eficacia de la seguridad del correo electrónico y la necesidad de evolucionar hacia enfoques de protección más avanzados.

Descripción:

El fuzzing se ha consolidado como una técnica de evasión relevante dentro de campañas modernas de spam y phishing a gran escala, empleada para introducir variaciones controladas

en elementos clave del correo electrónico con el objetivo explícito de degradar la eficacia de los sistemas de detección automatizados. Al adoptar esta técnica, los atacantes dejan de enviar correos idénticos y comienzan a utilizar plantillas dinámicas que cambian de forma controlada elementos clave del mensaje, como el asunto, el nombre del remitente y partes del contenido, sin modificar el objetivo del ataque.

Desde el punto de vista operativo, el fuzzing no consiste en generar cambios al azar, sino en aplicar variaciones cuidadosamente definidas. Los atacantes introducen cambios siguiendo reglas claras, como usar textos de la misma longitud, conjuntos limitados de caracteres o frases en las que solo se modifica una parte concreta. Este control les permite ajustar el nivel de variación según el comportamiento de los filtros de correo —por ejemplo, si los mensajes se entregan correctamente o son bloqueados— y así evitar activar de forma consistente los mecanismos de detección.

Para los sistemas de seguridad, este comportamiento provoca que los indicadores del ataque se dispersen intencionalmente. Aunque los correos persiguen el mismo objetivo malicioso, las pequeñas diferencias en su redacción son suficientes para impedir que los sistemas tradicionales los identifiquen y agrupen como parte de una misma campaña. Como resultado, los ataques aparecen como eventos aislados, lo que dificulta su detección temprana y el reconocimiento de una operación coordinada.

El impacto de esta técnica es significativo en la operación diaria de los equipos de seguridad, ya que reduce la visibilidad de las campañas de correo malicioso y retrasa la generación de alertas consolidadas en los entornos SOC. En lugar de detectar una campaña completa, los equipos deben analizar correos sospechosos de forma individual, lo que aumenta la carga de trabajo y alarga los tiempos de respuesta. Este contexto evidencia las limitaciones de los sistemas de protección del correo electrónico basados únicamente en indicadores estáticos y

subraya la necesidad de adoptar enfoques más avanzados, centrados en el análisis de comportamiento y la evaluación contextual del contenido.

Consecuencias de la explotación

La explotación efectiva de técnicas de *fuzzing* en campañas de spam y phishing puede generar impactos relevantes tanto a nivel técnico como operativo para las organizaciones afectadas.

Entre las principales consecuencias se destacan las siguientes:

- **Pérdida de datos:** Explotar una vulnerabilidad puede resultar en la pérdida, alteración o destrucción de datos valiosos, lo que compromete la integridad de la información crítica de la organización, dificultando la recuperación e impactando directamente en la operación.

- **Acceso no autorizado**: Los atacantes pueden explotar vulnerabilidades para obtener acceso no autorizado a sistemas y datos sensibles, lo que pone en riesgo la privacidad y seguridad de la información confidencial, permitiendo el uso indebido de recursos internos y exponiendo la organización a futuros ataques.

- **Robo de identidad:** La explotación de vulnerabilidades puede resultar en la exposición de datos confidenciales, lo que afecta gravemente la confianza de los clientes y socios, dañando la reputación de la organización y afectando su capacidad para atraer y retener clientes.

- **Daño económico:** La explotación de vulnerabilidades críticas puede interrumpir las operaciones comerciales, afectando la disponibilidad de los servicios y productos clave, lo que provoca un impacto negativo en la productividad, la cadena de suministro y la capacidad de la empresa para cumplir con sus compromisos.

- **Compromiso de sistemas y dispositivos conectados**: Los atacantes pueden explotar vulnerabilidades en dispositivos conectados a la red para comprometer otros sistemas de la infraestructura organizacional, creando una red de ataque que se propaga rápidamente, afectando la seguridad general de la organización.

• **Uso indebido de recursos:** A través de la explotación de vulnerabilidades, los atacantes pueden utilizar los recursos computacionales de la organización para actividades maliciosas, como el envío de spam o la minería de criptomonedas, lo que genera un aumento en los costos operativos y un desvío significativo de recursos.

• **Propagación de malware:** La explotación de vulnerabilidades permite la introducción de malware en los sistemas comprometidos, lo que resulta en una propagación rápida de software malicioso a otros dispositivos y redes, afectando la seguridad general y la capacidad de la organización para protegerse frente a amenazas externas.

Modo de explotación del ataque

La cadena de ataque de *fuzzing* puede dividirse en cinco fases claramente identificadas, las cuales se describen a continuación en detalle:

Fase 1 – Introducción de entradas aleatorias o malformadas

El ataque comienza con el envío de entradas malformadas o aleatorias a la aplicación o sistema objetivo. En el caso de un ataque de fuzzing, el atacante no necesita conocer los detalles internos del sistema, sino que se enfoca en provocar comportamientos inesperados a través de datos inesperados. Estas entradas pueden ser cadenas de texto, números o datos estructurados, con el objetivo de que el sistema falle o se comporte de manera anómala. El fuzzing puede ser realizado de manera automatizada, lo que aumenta significativamente la cantidad de entradas posibles.

Fase 2 – Identificación de vulnerabilidades y evaluación de la respuesta

Cuando el atacante envía información incorrecta al sistema, está observando cómo responde para detectar posibles fallos o debilidades. La evaluación de las respuestas permite identificar vulnerabilidades potenciales, como desbordamientos de búfer, errores de validación de datos o fallos en el manejo de excepciones. Esta fase es crucial para determinar qué entradas

provocan comportamientos peligrosos en la aplicación, lo que puede conducir a la explotación de esas vulnerabilidades.

Fase 3 – Evasión de controles de seguridad tradicionales

En un ataque de fuzzing aplicado al correo electrónico, por ejemplo, el atacante puede variar de forma controlada elementos clave dentro de los mensajes (como el nombre del remitente, asunto o contenido) para eludir los mecanismos de detección de spam o phishing. Las entradas generadas por el fuzzing son suficientemente diferentes entre sí para evitar que los sistemas de detección los agrupe bajo un mismo patrón, lo que hace más difícil que sean identificados por filtros tradicionales basados en firmas o reglas de coincidencia exacta. Esta fase aprovecha la variabilidad controlada para mantenerse por debajo del umbral de detección de sistemas de seguridad.

Fase 4 – Persistencia y ajuste dinámico

El atacante ajusta dinámicamente las entradas generadas por el fuzzer en función de las respuestas obtenidas durante la fase de evaluación. Este ajuste dinámico permite al atacante modificar las características de las entradas para mantener el ataque en curso sin ser detectado. Por ejemplo, si un sistema empieza a identificar un patrón específico en las entradas malformadas, el atacante puede modificar ligeramente los datos para eludir la detección, manteniendo así la persistencia del ataque a lo largo del tiempo.

Fase 5 – Explotación de la vulnerabilidad descubierta

Una vez que se ha identificado una vulnerabilidad a través del fuzzing, el atacante puede explotar esa vulnerabilidad para ejecutar código arbitrario, acceder a información sensible o incluso comprometer completamente el sistema. En algunos casos, el atacante puede utilizar los resultados del fuzzing para automatizar la explotación, utilizando los mismos patrones de

entradas que provocaron el fallo para escalar el ataque y obtener el control completo del sistema o red comprometida.

Vulnerabilidades asociadas

El fuzzing es una técnica que permite descubrir vulnerabilidades mediante la inyección de entradas malformadas o aleatorias a un sistema. En este informe, se detallan las vulnerabilidades asociadas al fuzzing, tanto en el contexto de correo electrónico (en ataques de spam y phishing) como en el contexto clásico de pruebas de seguridad de software.

- **Dependencia de mecanismos de detección estáticos:** al introducir variaciones controladas en asuntos, alias, nombres de remitente y otros atributos del mensaje, la campaña puede eludir filtros basados en firmas, reputación y coincidencias exactas, manteniéndose por debajo de los umbrales de detección y dificultando su identificación como una sola operación maliciosa.
- **Deficiencias en la correlación y agrupación de campañas:** cuando la plataforma de seguridad no relaciona mensajes por múltiples atributos y contexto, la personalización, el morphing (técnica que crea una transición gradual entre dos imágenes o formas) y otras técnicas de evasión dificultan reconocer que varios correos pertenecen al mismo ataque, retrasando la respuesta y la remediación. Microsoft señala que el análisis moderno debe identificar correos relacionados porque los ataques rara vez consisten en un único mensaje y porque los atacantes “morph” parámetros del correo para evitar detección.
- **Cobertura insuficiente frente a spoofing e impersonación:** el fuzzing en correo se vuelve más eficaz cuando la organización no tiene controles sólidos contra remitentes falsificados o suplantación de usuarios y dominios. Microsoft indica que las políticas anti-phishing están diseñados precisamente para detectar spoofing, impersonación y otras técnicas engañosas de correo.

- **Autenticación débil o incompleta del correo electrónico:** la ausencia o mala implementación de SPF, DKIM y DMARC facilita que mensajes fraudulentos aparenten legitimidad y lleguen al usuario con menos fricción. CISA explica que DMARC funciona junto con SPF y DKIM para autenticar remitentes y ayudar a los sistemas receptores a validar mensajes enviados desde un dominio.
- **Insuficiente análisis heurístico y conductual:** Hornetsecurity y Microsoft coinciden en que los controles modernos no deben depender solo de firmas; deben incorporar heurística, comportamiento, IA y análisis de campañas coordinadas para detectar amenazas nuevas o modificadas.
- **Alta exposición del usuario final ante correos sofisticados:** a medida que los mensajes se ven menos repetitivos y más naturales, aumenta la probabilidad de que el usuario los perciba como legítimos. Microsoft advierte que, con la creciente complejidad de los ataques, incluso usuarios entrenados tienen dificultades para identificar mensajes sofisticados de phishing.

Ahora bien, en relación con el tema del fuzzing clásico de software, estas son las vulnerabilidades técnicas que más suele descubrir:

- **Desbordamientos de búfer y escrituras fuera de límites:** OWASP indica que el fuzzing busca fallos de implementación y puede llevar a problemas clásicos como buffer overflows y DoS; MITRE agrega que una escritura fuera de límites puede causar corrupción de memoria, caída del proceso o ejecución de código no autorizado.
- **Lecturas fuera de límites y sobrelecturas de memoria:** MITRE describe las lecturas fuera de límites como fallas en cálculos de longitud, tamaño de buffer u offset, y NVD muestra que este tipo de errores puede desembocar en denegación de servicio mediante solicitudes especialmente construidas.

- **Use-after-free y double free:** Microsoft y MITRE incluyen entre las fallas típicas detectables errores como use-after-free y double free, que comprometen la estabilidad y, en ciertos contextos, pueden habilitar explotación más grave.
- **Integer overflow y cadenas de fallas derivadas:** MITRE señala que un integer overflow puede desencadenar corrupción de memoria o incluso buffer overflows posteriores, especialmente cuando afecta cálculos de reserva de memoria.
- **Null pointer dereference y fallas de disponibilidad:** Microsoft incluye las null pointer dereferences entre los errores de memoria que estas técnicas ayudan a diagnosticar, normalmente con impacto en estabilidad y disponibilidad del servicio.

Recomendaciones de detección y mitigación

Los equipos de SOC deben configurar reglas de detección basadas en análisis conductual en plataformas SIEM y EDR para identificar patrones de fuzzing en correos electrónicos. Es fundamental implementar correlación de eventos, análisis de variabilidad en atributos clave (remitente, asunto) y monitoreo de comportamiento de entrega para detectar evasión de filtros tradicionales.

Implementar Filtros de Seguridad Basados en Análisis de Comportamiento:

1. Analizar patrones de tráfico: Implementar soluciones que inspeccionen el comportamiento de los correos electrónicos en busca de variaciones sutiles y patrones de evasión que los filtros tradicionales no pueden identificar.
2. Evaluación de la actividad del remitente: Monitorizar la frecuencia y tipo de variación de los atributos del correo (como alias, asunto, nombre del remitente) para identificar posibles intentos de evasión.

3. Herramientas de análisis de contexto: Utilizar tecnologías avanzadas de IA y machine learning que puedan detectar comportamientos anómalos o desconocidos, especialmente en campañas de phishing o spam.

Fortalecer la Autenticación de Correo Electrónico (SPF, DKIM, DMARC)

1. Configurar y monitorear SPF, DKIM y DMARC: Asegurarse de que estos estándares estén configurados correctamente para verificar la legitimidad de los remitentes.
2. Aplicar políticas estrictas de autenticación: Hacer uso de políticas DMARC para rechazar los correos electrónicos que no pasen la verificación de autenticidad.
3. Alertas en caso de errores de autenticación: Establecer alertas automáticas si un correo no pasa las verificaciones SPF/DKIM, lo que podría indicar un intento de suplantación de identidad.

Implementar Sistemas de Detección de Phishing Adaptativos

1. Análisis de enlaces y contenido: Desarrollar sistemas que no solo verifiquen la reputación de los enlaces, sino que también analicen el contexto del mensaje y la coherencia del contenido.
2. Entrenamiento continuo: Utilizar plataformas de IA que aprendan constantemente sobre nuevas técnicas de phishing y ataques de fuzzing, para actualizar los filtros y detectar nuevas variaciones.
3. Simulaciones regulares de phishing: Realizar pruebas continuas de phishing internas para evaluar cómo los empleados reaccionan ante correos electrónicos falsificados o modificados.

Recomendaciones de prevención y mitigación

En paralelo a la detección, se recomienda aplicar las siguientes medidas de fortalecimiento y prevención:

Uso de Herramientas de Fuzzing en las Etapas de Desarrollo

1. Integrar fuzzing en el CI/CD: Incorporar herramientas de fuzzing en los pipelines de integración y despliegue continuo (CI/CD) para realizar pruebas automatizadas durante las etapas de desarrollo.
2. Uso de fuzzers avanzados: Adoptar herramientas de fuzzing como AFL (American Fuzzy Lop), LibFuzzer o Peach Fuzzer para detectar vulnerabilidades como desbordamientos de búfer y errores de validación de entrada.
3. Automatizar análisis de seguridad: Incorporar fuzzing junto con otras técnicas de análisis estático y dinámico para maximizar la cobertura y detectar vulnerabilidades de manera temprana.

Configuración Adecuada de Protección de Memoria

1. Activar protecciones de memoria: Asegurarse de que las configuraciones del sistema operativos y aplicaciones incluyan protecciones de seguridad como DEP (Data Execution Prevention) y ASLR (Address Space Layout Randomization).
2. Uso de herramientas de sanitización: Implementar herramientas como AddressSanitizer o MemorySanitizer para detectar errores en la memoria durante las pruebas de fuzzing y prevenir ataques de use-after-free o double free.
3. Revisar regularmente el código: Auditar de manera regular el código fuente para identificar prácticas de manejo inseguro de la memoria, como buffer overflows, que pueden ser detectadas mediante fuzzing.

Implementar Reglas de Seguridad Basadas en Heurística y Análisis Comportamental

1. Utilizar herramientas de análisis heurístico: Emplear soluciones que realicen análisis heurístico para detectar anomalías y patrones no conocidos, más allá de los filtros tradicionales basados en firmas.

2. Implementar sistemas de análisis de comportamiento: Configurar sistemas que analicen el comportamiento de las aplicaciones y los usuarios, buscando comportamientos atípicos que podrían ser indicativos de un ataque basado en fuzzing.
3. Desarrollar sistemas de aprendizaje automático: Implementar modelos de IA y machine learning que puedan aprender de las campañas de ataques pasadas y adaptarse para detectar nuevos ataques de fuzzing.

Reforzar los Controles de Acceso y Privilegios

1. Aplicar el principio de privilegio mínimo: Limitar los privilegios de los usuarios y aplicaciones para minimizar el impacto de un ataque.
2. Revisión de roles y permisos: Auditar periódicamente los roles y permisos dentro de los sistemas para asegurarse de que solo los usuarios autorizados tengan acceso a recursos críticos.
3. Autenticación multifactor (MFA): Reforzar el acceso a sistemas y aplicaciones mediante MFA para dificultar el acceso no autorizado.

Recomendaciones de respuesta ante compromiso

En caso de confirmarse o sospecharse compromiso por SpankRAT, se recomienda ejecutar el siguiente procedimiento de contención y erradicación:

1. Aislar las cuentas y sistemas afectados que han recibido correos maliciosos para evitar la propagación de la amenaza. Esto incluye la desconexión de servidores de correo comprometidos y la detención de la transmisión de correos sospechosos.
2. Revocar las credenciales de las cuentas comprometidas para evitar el acceso no autorizado. Esto debe ir acompañado de la actualización de contraseñas y la implementación de autenticación multifactor (MFA) en todas las cuentas de correo empresarial.

3. Ajustar las reglas de filtrado de correo electrónico en plataformas como Microsoft 365, Secure Email Gateway, y SIEM para detectar patrones de fuzzing. Crear reglas de respuesta automática para bloquear correos similares con atributos o metadatos coincidentes con las campañas maliciosas.
4. Monitorear las actividades de correo electrónico de forma continua utilizando herramientas de análisis comportamental. Además, realizar simulaciones de phishing basadas en fuzzing para entrenar a los empleados a detectar técnicas de evasión.
5. Revisar los logs del sistema para identificar el origen del ataque y los patrones en las entradas malformadas que fueron inyectadas. Analizar el tráfico de red para buscar solicitudes HTTP o WebSocket sospechosas relacionadas con el fuzzing.
6. Revertir cualquier cambio sospechoso realizado por el atacante, como modificaciones en la configuración del sistema o archivos maliciosos inyectados. Aplicar parches de seguridad a todas las vulnerabilidades relacionadas con el fuzzing, como desbordamientos de búfer, errores de validación y acceso no autorizado.
7. Si el fuzzing se utilizó para explotar vulnerabilidades específicas (por ejemplo, desbordamientos de búfer o inyecciones de código), desactivar temporalmente los componentes vulnerables mientras se investiga más a fondo.
8. Habilitar protecciones de memoria como DEP (Data Execution Prevention) y ASLR (Address Space Layout Randomization) en todos los sistemas afectados. Además, implementar herramientas de fuzzing automatizado en el proceso de desarrollo para detectar vulnerabilidades en futuras versiones de software.
9. Realizar una evaluación de seguridad exhaustiva del sistema para identificar otras posibles vulnerabilidades que podrían haber sido pasadas por alto. Esto debe incluir pruebas de fuzzing adicionales sobre nuevas versiones del software, así como un análisis en profundidad de la arquitectura de seguridad.

10. Implementar monitoreo continuo a través de soluciones SIEM y EDR, enfocándose en la detección de entradas no válidas y comportamientos inusuales generados por fuzzing. También se debe realizar un ciclo de retroalimentación para mejorar las capacidades de detección y mitigación frente a ataques de fuzzing.
11. Implementar soluciones basadas en IA para detectar patrones de fuzzing, especialmente aquellas que se enfocan en el análisis de comportamiento y la identificación de entradas anómalas.
12. Establecer reglas de detección avanzadas para identificar variaciones sutiles en entradas o mensajes que puedan estar siendo manipuladas por fuzzing.
13. Entrenar regularmente a los equipos de seguridad y usuarios finales sobre los riesgos del fuzzing y otras técnicas de evasión. Proporcionar escenarios prácticos sobre cómo identificar correos electrónicos sospechosos o respuestas erróneas de los sistemas.
14. Reforzar los controles de acceso a sistemas críticos mediante políticas de privilegio mínimo y autenticación multifactor (MFA) para reducir las oportunidades de explotación por parte de atacantes.
15. Aplicar actualizaciones de seguridad de manera continua y proactiva para abordar las vulnerabilidades que los atacantes puedan explotar mediante fuzzing.

Conclusiones

El fuzzing es una técnica avanzada que, aunque se utiliza en pruebas de seguridad, también puede ser explotada por atacantes para evadir los controles tradicionales de seguridad. La respuesta efectiva ante un compromiso de fuzzing debe centrarse en aislar y mitigar los efectos del ataque, mientras se implementan estrategias de monitoreo y detección avanzadas. Con la preparación adecuada y una respuesta rápida, las organizaciones pueden reducir el impacto de

este tipo de ataque y fortalecer su postura de seguridad frente a futuros intentos de explotación.

Fuentes:

- Hornetsecurity Monthly Threat Report. Marzo de 2026. Disponible en: Hornetsecurity Monthly Threat Report Marzo 2026. y https://www.hornetsecurity.com/es/blog/monthly-threat-report-marzo-2026/
- AFL (American Fuzzy Lop)– https://lcamtuf.coredump.cx/afl/.
- MITRE ATT&CK Framework for Enterprise– https://attack.mitre.org/.

En caso de dudas, inquietudes o requerir asistencia adicional relacionada con esta alerta de seguridad, puede comunicarse directamente con el **CSIRT Salud** a través de las líneas telefónicas **(+57) 316 893 1490 - 318 155 3570** o mediante el correo electrónico **csirtsalud@minsalud.gov.co**. Nuestro equipo está disponible para brindar el acompañamiento necesario.