Alerta ID:	095
Fecha del reporte:	23/04/2026
Entidad:	Todas las entidades del ecosistema digital
Título:	Malware GoGra para Linux.
Herramienta de detección	N/A
Activo involucrado:	Sistemas Linux
Tipo de alerta:	Alerta
Nivel de riesgo:	Alto

Objetivo:

Informar a las entidades del ecosistema digital sobre la identificación de una nueva amenaza dirigida a sistemas Linux, denominada GoGra. Este malware utiliza la API de Microsoft Graph para establecer comunicaciones con los atacantes, lo que le permite evadir las soluciones de detección tradicionales basadas en reputación. La amenaza emplea técnicas avanzadas para ocultar su actividad dentro de procesos legítimos del sistema operativo, lo que aumenta su capacidad de persistencia y reduce la visibilidad ante los controles de seguridad, convirtiéndolo en un riesgo significativo para las infraestructuras comprometidas.

Descripción:

GoGra es una nueva variante de malware dirigido a sistemas Linux, identificado por su capacidad para explotar la API de Microsoft Graph en su comunicación con los atacantes. Este

malware ha sido diseñado para evitar la detección tradicional y mantener un canal oculto de comando y control (C2), lo que lo convierte en una amenaza difícil de rastrear.

A través de un enfoque innovador, GoGra utiliza técnicas de evasión avanzadas, especialmente al enmascarar sus comunicaciones dentro de procesos legítimos del sistema operativo Linux. El malware se infiltra dentro de estos procesos, lo que le permite permanecer activo sin generar alertas en los sistemas de detección basados en reputación o firmas. Una de sus características distintivas es el uso de la API de Microsoft Graph, que permite que el malware se comunique con los atacantes de forma encubierta, aprovechando una infraestructura legítima para evitar la inspección del tráfico de red.

El análisis realizado por los expertos en seguridad ha demostrado que el malware tiene la capacidad de mantener persistencia en el sistema afectado y de ejecutar comandos de forma remota, sin ser detectado fácilmente. Esta amenaza representa un riesgo significativo para las organizaciones, ya que su capacidad de infiltración y evasión puede ser utilizada para acceder a información sensible o comprometer sistemas críticos.

Consecuencias de la explotación

La explotación exitosa de **GoGra** puede tener repercusiones graves para las entidades afectadas, tanto a nivel técnico como operativo. Entre las principales consecuencias se destacan las siguientes:

1. **Compromiso total del endpoint**: Los atacantes obtienen control remoto completo sobre el sistema infectado, permitiendo la ejecución de comandos arbitrarios, la manipulación de procesos y la obtención de datos sensibles.
2. **Evasión prolongada de detección**: Gracias a su capacidad para operar dentro de procesos legítimos del sistema, como el uso de la **API de Microsoft Graph**, **GoGra** es

capaz de eludir los controles de detección basados en reputación, permaneciendo oculto durante un largo periodo antes de ser identificado.

3. **Persistencia resistente**: El malware puede mantenerse activo a través de técnicas de persistencia, lo que asegura que el sistema comprometido siga siendo accesible para los atacantes incluso tras reinicios o interrupciones en su funcionamiento.
4. **Exfiltración de datos sensibles**: **GoGra** tiene la capacidad de acceder a información confidencial de la entidad afectada, incluyendo credenciales, datos financieros o información personal de los usuarios, lo que puede derivar en violaciones de privacidad y daños a la reputación.
5. **Riesgos operacionales**: Al mantener el control del sistema comprometido, los atacantes pueden interrumpir servicios, realizar manipulaciones de configuraciones críticas o causar fallos operativos que afecten a la disponibilidad de los servicios de la entidad.
6. **Impacto en la infraestructura de TI**: La presencia prolongada del malware en los sistemas puede llevar a un desgaste de los recursos tecnológicos, incrementar la carga de trabajo para los equipos de seguridad y exigir una intervención prolongada para la remediación.

Modo de explotación del ataque

La cadena de ataque de GoGra se desarrolla en varias fases claramente definidas, las cuales describimos a continuación:

Fase 1 – Entrega y ejecución inicial

El ataque comienza con la entrega e instalación del malware a través de un vector de ataque que utiliza exploit kits o ingeniería social para inducir a las víctimas a ejecutar el malware. Una vez ejecutado, el GoGra se instala como un proceso legítimo dentro del sistema operativo, camuflándose para evitar la detección temprana.

Fase 2 – Comunicación con los atacantes mediante la API de Microsoft Graph

Una vez ejecutado en el sistema, GoGra establece una comunicación con los atacantes a través de la API de Microsoft Graph. Utilizando esta API, el malware enmascara sus comunicaciones dentro del tráfico legítimo de la red, lo que hace más difícil su detección por soluciones de seguridad tradicionales basadas en la inspección de tráfico.

Fase 3 – Persistencia y evasión de detección

El malware asegura su persistencia mediante técnicas avanzadas de evasión. Se oculta dentro de procesos legítimos del sistema, como explorer.exe, lo que le permite operar sin generar alertas de seguridad, ya que el tráfico malicioso parece provenir de un proceso confiable y firmado por el sistema operativo.

Fase 4 – Ejecución remota de comandos

Una vez que el malware se ha infiltrado y establecido la comunicación con el servidor de comando y control (C2), el atacante puede ejecutar comandos de forma remota en el sistema comprometido. Esto incluye desde la ejecución de archivos y scripts maliciosos hasta la manipulación de registros, servicios y procesos críticos del sistema operativo.

Fase 5 – Exfiltración de datos y movimiento lateral

En esta fase, GoGra puede ser utilizado para exfiltrar información sensible, como credenciales o datos financieros. Además, el malware puede facilitar el movimiento lateral dentro de la red de la víctima, buscando otros sistemas susceptibles de ser comprometidos.

Vulnerabilidades asociadas

A diferencia de otras amenazas que explotan vulnerabilidades específicas con identificadores CVE, GoGra no se apoya en la explotación de una vulnerabilidad conocida o publicada. Su efectividad radica en el abuso de características legítimas del sistema operativo Linux y en

debilidades comunes en las configuraciones y visibilidad de las redes corporativas. Las principales técnicas aprovechadas por este malware incluyen:

- **Uso indebido de la API de Microsoft Graph:** GoGra abusa de la infraestructura legítima proporcionada por Microsoft, específicamente la API de Microsoft Graph, para establecer un canal de comunicación encubierto con los atacantes. Este abuso de una herramienta confiable dificulta la detección del tráfico malicioso, al enmascararlo dentro de comunicaciones legítimas.
- **Evasión mediante la inyección de código en procesos de confianza:** Utilizando una técnica clásica de inyección de DLL en procesos legítimos como explorer.exe, el malware puede ejecutar código malicioso dentro de procesos de sistema operativo firmados y confiables, lo que permite evitar los controles de seguridad basados en reputación.
- **Falta de controles adecuados en la configuración de seguridad:** La ausencia de controles estrictos sobre la ejecución de scripts y la ejecución de PowerShell, como el uso del parámetro -ExecutionPolicy Bypass, permite que GoGra eluda políticas de seguridad establecidas para bloquear la ejecución de código no autorizado.
- **Explotación de privilegios del sistema operativo:** El malware abusa de la configuración de SeDebugPrivilege, un privilegio legítimo en el sistema operativo, que permite al malware obtener acceso a procesos protegidos y realizar inyección de código en procesos de mayor privilegio. Esta debilidad no es una vulnerabilidad específica, sino un abuso de una característica legítima que, si no se gestiona correctamente, permite a los atacantes comprometer la seguridad del sistema.
- **Debilidad en la inspección de tráfico saliente:** El tráfico WebSocket malicioso enviado por GoGra se comunica a través de puertos no estándar (como el 9000/TCP), lo que suele ser ignorado por las soluciones de seguridad perimetrales. Esto permite que las comunicaciones del malware eviten la inspección y permanezcan ocultas.

- **Falta de visibilidad en la gestión de tareas programadas:** GoGra explota la capacidad del Programador de tareas de Windows para crear tareas con privilegios elevados, algo que podría ser evitado con controles de seguridad más estrictos. Sin una adecuada supervisión, los atacantes pueden programar tareas que se ejecuten en cada inicio de sesión, garantizando la persistencia del malware.

Recomendaciones de detección y mitigación

Para detectar y mitigar eficazmente la amenaza representada por **GoGra**, es fundamental que las entidades implementen medidas de seguridad avanzadas, orientadas a la identificación de comportamientos maliciosos y a la protección de sus sistemas. A continuación, se presentan algunas recomendaciones clave:

1. **Monitoreo de inyecciones de DLL sobre procesos legítimos:** Implementar reglas de detección que alerten sobre la inyección de DLLs maliciosas en procesos legítimos del sistema, como explorer.exe. Estas inyecciones suelen realizarse desde rutas no confiables, como **C:\ProgramData**, y deben ser identificadas a través de análisis detallados de los procesos en ejecución.
2. **Detección de creación de tareas programadas con privilegios elevados:** Establecer alertas sobre la creación de tareas programadas que se ejecuten con el nivel de privilegios más alto (highest privileges) y que estén configuradas para ejecutarse en el inicio de sesión del usuario. Las tareas relacionadas con RmmAgentCore son un indicador claro de actividad maliciosa.
3. **Monitoreo de conexiones WebSocket salientes:** Monitorear las conexiones de salida hacia puertos no estándar (como el 9000/TCP) utilizando WebSocket, que es el método de comunicación preferido por GoGra para contactar con sus servidores de comando y control (C2). Es esencial inspeccionar cualquier tráfico que se desvíe de las rutas de comunicación estándar.

4. **Análisis de solicitudes HTTP sospechosas:** Identificar patrones en las solicitudes HTTP que incluyan URLs características de descarga de payloads maliciosos, como /download/rmm_agent.dll, a través de registros de proxies, firewalls y telemetría de plataformas EDR/IDS.
5. **Revisión de ejecuciones de PowerShell sospechosas:** Alertar sobre ejecuciones de PowerShell con parámetros -NoProfile, -NonInteractive y -ExecutionPolicy Bypass, especialmente cuando son invocadas por procesos legítimos como explorer.exe. Esto puede indicar que el malware está utilizando PowerShell para ejecutar comandos maliciosos sin ser detectado.
6. **Supervisión de escalamiento de privilegios:** Monitorear el uso del privilegio SeDebugPrivilege y las acciones asociadas al comando Start-Process -Verb RunAs. Estos son métodos que los atacantes pueden utilizar para escalar privilegios y ejecutar código malicioso en procesos del sistema.
7. **Integración de inteligencia de amenazas:** Incorporar los Indicadores de Compromiso (IoC) identificados para GoGra, como direcciones IP, hashes de archivos y patrones de tráfico en las plataformas SIEM, EDR y IDS/IPS para detectar patrones de ataque y prevenir infecciones.

Recomendaciones de prevención y mitigación

Para reducir el riesgo de infección por **GoGra** y fortalecer la seguridad de los sistemas ante este tipo de amenazas, se recomienda aplicar las siguientes medidas preventivas:

1. **Restricción del privilegio SeDebugPrivilege**: Limitar el uso del privilegio SeDebugPrivilege exclusivamente a cuentas de administradores controladas mediante políticas de grupo (GPO). Se debe monitorear continuamente la asignación de este privilegio para evitar que usuarios no autorizados o procesos maliciosos lo utilicen para escalar privilegios y realizar inyecciones de código.
2. **Implementación de AppLocker o Windows Defender Application Control (WDAC):** Configurar AppLocker o WDAC para bloquear la ejecución de binarios y DLLs no

firmados o provenientes de ubicaciones no autorizadas. Esto limitará la ejecución de código malicioso, como el utilizado por GoGra, en entornos corporativos.

3. **Activación de Constrained Language Mode en PowerShell**: Habilitar el Constrained Language Mode en PowerShell para restringir las capacidades de ejecución de scripts y módulos no autorizados. Además, activar la auditoría de scripts, mediante ScriptBlock Logging y Module Logging, permitirá detectar intentos de ejecución maliciosa a través de PowerShell.
4. **Bloqueo de ejecución de binarios en rutas sensibles**: Aplicar controles para bloquear la ejecución de archivos desde rutas sensibles y no autorizadas, como **C:\ProgramData**, *C:*Users**Public** o perfiles de usuario no administrativos. Esto evitará que el malware se ejecute desde ubicaciones fuera de los directorios de programas estándar.
5. **Monitoreo de conexiones WebSocket no autorizadas**: Configurar el firewall y las soluciones de inspección de tráfico para bloquear o monitorear las conexiones WebSocket (ws://) hacia puertos no estándar, como el 9000/TCP, para detectar y prevenir la comunicación del malware con los servidores de comando y control (C2).
6. **Inspección profunda de tráfico HTTP saliente**: Implementar inspección profunda de paquetes (DPI) para bloquear solicitudes HTTP salientes hacia servidores de C2 con reputación desconocida. Esto evitará que los binarios maliciosos sean descargados y ejecutados en el sistema.
7. **Fortalecimiento del Programador de Tareas de Windows**: Limitar la capacidad de crear tareas programadas con privilegios elevados a cuentas administrativas auditadas. La creación de tareas programadas maliciosas, como las que utiliza GoGra para asegurar la persistencia, debe ser restringida mediante políticas de seguridad.
8. **Implementación de soluciones de análisis dinámico de malware**: Utilizar soluciones de análisis dinámico, como ANY.RUN, Joe Sandbox o Cuckoo, en los flujos de triage del

equipo de seguridad para detectar amenazas que evaden la detección basada en firmas, como GoGra.

9. **Educación y concienciación sobre seguridad cibernética**: Fomentar la educación y concienciación en las entidades sobre los riesgos asociados con malware como GoGra y las mejores prácticas para evitar la ejecución de programas desconocidos o no confiables, especialmente a través de ingeniería social

Recomendaciones de respuesta ante compromiso

En caso de confirmarse o sospecharse un compromiso por **GoGra**, se recomienda ejecutar el siguiente procedimiento de **contención** y **erradicación** para minimizar los daños y recuperar el control del sistema afectado:

1. **Aislar el equipo afectado**: Desconectar el equipo comprometido de la red inmediatamente, ya sea de forma lógica (mediante plataformas EDR) o física. Esto previene la propagación del malware a otros sistemas y asegura la preservación de la evidencia volátil, como la memoria RAM y los artefactos del sistema.
2. **Captura de evidencia forense**: Antes de tomar cualquier acción correctiva, capturar evidencia crítica, como:
    - Imagen del disco y logs de eventos de Windows.
    - Logs de PowerShell y cualquier tarea programada relacionada con RmmAgentCore.
    - Información sobre conexiones WebSocket y comunicaciones con servidores de comando y control (C2).
3. **Eliminar tareas programadas maliciosas**: Verificar y eliminar cualquier tarea programada creada por GoGra, como la tarea RmmAgentCore. Asegúrese de que no queden tareas con privilegios elevados o configuradas para ejecutarse al inicio de sesión.

4. **Eliminar archivos maliciosos**: Eliminar los archivos identificados como maliciosos, como rmm_agent.dll, RmmAgentCore.exe y arc_agent.exe, de las rutas en las que hayan sido desplegados, como *C:\ProgramData* u otras ubicaciones sospechosas.
5. **Verificación de inyecciones en explorer.exe**: Utilizar herramientas forenses como Volatility, Process Hacker o PE-sieve para verificar la existencia de inyecciones de código en explorer.exe. Si se detectan, proceder a su eliminación y reiniciar el sistema para garantizar la limpieza.
6. **Restablecimiento de credenciales**: Cambiar todas las credenciales de usuarios que hayan iniciado sesión en el sistema comprometido. Además, revisar los secretos y tokens expuestos para asegurarse de que no hayan sido comprometidos por los atacantes.
7. **Revisión de comunicación saliente**: Inspeccionar las comunicaciones salientes hacia las direcciones IP de los servidores de C2, como 45.131.214[.]132 y 166.1.144[.]109 en el puerto 9000/TCP. Esto permitirá identificar si hay otros equipos comprometidos dentro de la red y determinar el alcance del ataque.
8. **Notificación a las autoridades**: Reportar el incidente al CSIRT Salud y, si corresponde, a colCERT y a la Superintendencia de Industria y Comercio, especialmente si hay afectación de datos personales de pacientes, conforme a la Ley 1581 de 2012 de Protección de Datos Personales.
9. **Reinstalación del sistema operativo**: En casos donde el compromiso sea prolongado o no se pueda validar la integridad completa del sistema afectado, considerar la reinstalación completa del sistema operativo para garantizar que no queden rastros de la infección.

Conclusiones

GoGra representa una amenaza significativa para las infraestructuras Linux, especialmente debido a su capacidad para eludir la detección utilizando técnicas avanzadas de evasión como la inyección de código en procesos legítimos y el uso de la API de Microsoft Graph para la comunicación con los atacantes. Su comportamiento sigiloso y su capacidad de mantener la persistencia lo convierten en un riesgo elevado, que puede permanecer oculto durante largos períodos antes de ser detectado.

Las entidades deben ser conscientes de la complejidad de este ataque y de la importancia de implementar estrategias de defensa en profundidad que combinen detección conductual, monitoreo proactivo y fortalecimiento de la configuración de seguridad. La combinación de detección de anomalías en el tráfico y la implementación de políticas estrictas de control de acceso y privilegios son esenciales para reducir el riesgo de compromiso.

Si bien GoGra no depende de una vulnerabilidad pública conocida y explota debilidades en la gestión de privilegios y la visibilidad de las redes, las medidas preventivas recomendadas pueden ayudar a las organizaciones a protegerse eficazmente. En caso de compromiso, es crucial seguir las recomendaciones de respuesta ante incidentes para minimizar el impacto y evitar la propagación de la amenaza.

En resumen, GoGra subraya la necesidad de una estrategia de seguridad más robusta y dinámica que esté preparada para enfrentar amenazas cada vez más sofisticadas y difíciles de detectar.

Fuentes:

- **BleepingComputer** – *New GoGra malware for Linux uses Microsoft Graph API for communications*. Disponible en: https://www.bleepingcomputer.com/news/security/new-gogra-malware-for-linux-uses-microsoft-graph-api-for-comms/
- **BleepingComputer** – *Microsoft releases emergency security updates for critical ASP.NET flaw*. Disponible en: https://www.bleepingcomputer.com/news/microsoft/microsoft-releases-emergency-security-updates-for-critical-aspnet-flaw/

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.