INFORME DE INTELIGENCIA DE AMENAZAS · SÍNTESIS DEFENSIVA

Nightmare Eclipse

Anatomía de una campaña de ocho zero-days contra Microsoft Defender y BitLocker: una sola clase de vulnerabilidad, seis semanas, y un rencor con MSRC que terminó armando al ransomware.

› autor	fennek.org
› fecha	2026-07-07 (datos verificados al 7-jul-2026)
› clasificación	TLP:CLEAR — distribución libre
› alcance	análisis de fuentes públicas y de los PoC divulgados; mecanismos a nivel conceptual, sin código ofensivo
› serie completa	los análisis originales, con sus reglas y actualizaciones, en fennek.org

CONTENIDO

TLP:CLEAR Todos los CVE, indicadores y detecciones citados provienen de reporting abierto (NVD, MSRC, CISA, Huntress, Picus, Cyderes, GuardSix, entre otros); las fuentes están al final del documento. Este informe describe mecanismos a nivel conceptual con fines defensivos: no contiene pasos de explotación ni código ofensivo.

1 · Resumen ejecutivo

Entre el 7 de abril y el 10 de junio de 2026, un investigador que firma como **Nightmare-Eclipse** publicó **ocho exploits de prueba de concepto (PoC) funcionales** contra componentes de seguridad de Windows —sobre todo Microsoft Defender y BitLocker—. No los reportó por el cauce de divulgación coordinada: los publicó con código, **cronometrados para caer los días posteriores a cada Patch Tuesday**, garantizando semanas de exposición sin parche.

No es un grupo APT ni una operación patrocinada. Es **una sola persona en disputa con el Microsoft Security Response Center (MSRC)** que decidió convertir su enojo en armamento público. Y funcionó: Huntress confirmó que al menos **BlueHammer, RedSun y UnDefend** se usaron en intrusiones reales con **ransomware** como objetivo final, y CISA incorporó los CVE a su catálogo *Known Exploited Vulnerabilities* (KEV).

El hallazgo central de este análisis es que la campaña no son ocho vulnerabilidades independientes, sino, en su mayoría, **la misma clase de vulnerabilidad reencontrada por rutas de código distintas**: una condición de carrera *Time-of-Check to Time-of-Use* (TOCTOU) en operaciones de fichero privilegiadas de Defender, armada con *oplocks* y *junctions* de NTFS, y rematada —en varias piezas— por Windows Error Reporting como vehículo de ejecución como SYSTEM. Parchear un CVE no cierra la clase; el actor lo demostró pasando de `CVE-2026-33825` a `CVE-2026-41091` a `CVE-2026-50656` sin cambiar de idea, solo de puerta.

> **La lección de fondo.** No es «parchea». Es que **confiar en el control de seguridad nativo como si fuera infalible es un supuesto explotable**. Aquí el propio Defender —el componente que debía remediar el fichero malicioso— fue el vehículo de la escalada a SYSTEM. Cuando el antivirus escribe con privilegios de SYSTEM y no revalida la ruta, el antivirus es superficie de ataque.

2 · Alcance, metodología y nota del autor

Este documento es una **síntesis defensiva** de una campaña pública, escrita desde el lado azul. No relata un incidente propio: reconstruye, a partir de reporting abierto y del análisis de los PoC divulgados, cómo funcionan las técnicas y —sobre todo— **qué se ve cuando se ejecutan** y cómo mitigarlas. El objetivo es entender **clases de vulnerabilidad**, no reproducir exploits; cada mecanismo se describe a nivel conceptual y va acompañado de su contramedida.

> **Nota del autor.** Esta investigación nació de estudiar los PoC públicos de la campaña en un **laboratorio aislado y controlado**, con el fin de entender a fondo las clases de vulnerabilidad que explotan y traducirlas en ingeniería de detección y hardening. Publico el resultado por si le resulta útil a otros investigadores o profesionales de la seguridad para entender el mismo terreno. Lo que sigue es la parte publicable y defensiva de ese trabajo: describe el *porqué* y el *qué observar*, no el *cómo armar*.

Fuentes y verificación. Todo lo que aquí se afirma se contrastó contra NVD/MSRC, avisos de CISA y análisis de proveedores (Huntress, Picus, Cyderes, GuardSix, ThreatLocker, Eclypsium, SecurityWeek, entre otros). Dos precisiones que conviene tener presentes desde el inicio: **UnDefend es un denial-of-service** de Defender (no una «manipulación de firmas»), y el *named pipe* `\\.\pipe\RoguePlanet` es un **artefacto del PoC público**, no un indicador de intrusión real.

Vigencia. Los datos están verificados al **7 de julio de 2026**. La sección 12 resume el estado de parcheo a esa fecha; dos piezas (RoguePlanet y GreatXML) seguían sin parche al cierre.

3 · El actor: un rencor, no una bandera

El actor opera bajo varios alias —**Nightmare-Eclipse** (su handle de GitHub), **Chaotic Eclipse** (su blog), y variantes como *Dead Eclipse* o *MSNightmare*—. Se presenta como un investigador individual con una cuenta pendiente con Microsoft. En sus publicaciones afirma que «alguien violó nuestro acuerdo y me dejó sin hogar, sin nada» y que personal de MSRC le dijo «que iban a arruinarme la vida, y lo hicieron».

A diferencia del primer reporting, su perfil ya no es el de un externo enojado, sino el de un **insider**: el actor sostiene —y las crónicas posteriores lo recogen— que trabajó a tiempo completo en seguridad de Microsoft entre **septiembre de 2022 y junio de 2025**, que la empresa ignoró sus reportes internos, dejó de comunicarse y terminó **borrando la propia cuenta con la que enviaba los hallazgos**, dejándolo «con cero centavos» por su trabajo. Ese detalle importa técnicamente: explica el **conocimiento profundo de los componentes internos** de Defender y WinRE que exhiben los PoC.

El patrón tiene precedente. En 2018–2019, **SandboxEscaper** hizo casi lo mismo: soltó una tanda de LPEs de Windows con código funcional, sin coordinar, en ruptura abierta con MSRC. La lección se repite: **un investigador enfadado con acceso a vulnerabilidades de calidad es, en la práctica, un proveedor de zero-days para cualquiera que sepa copiar y pegar** —incluidas bandas de ransomware que jamás las habrían encontrado por su cuenta—.

El detalle operativo que hace tanto daño es el **timing**. Publicar el PoC *justo después* del Patch Tuesday no es casualidad: maximiza la ventana en la que la vulnerabilidad ya es pública pero todavía no hay parche. Es divulgación diseñada para causar el mayor daño posible dentro de un ciclo de parcheo.

> **Actualización · 7-jul-2026.** Tras publicar RoguePlanet, el actor amenazó con un «bone shattering drop» —una **divulgación masiva de zero-days** para el **14 de julio**—. La semana del 7 de julio la canceló: dijo que RoguePlanet lo dejó «drenado», que quizá se tome un descanso, y se disculpó por el pánico generado. La amenaza queda latente, no anulada; pero por ahora no hay dump masivo previsto. Que la fatiga de una sola persona sea lo que frena la próxima oleada dice mucho sobre la naturaleza de esta «campaña».

4 · La campaña de un vistazo

Ocho piezas, tres oleadas. Los nombres los puso el propio actor. La última columna refleja el estado al cierre de este informe (7-jul-2026).

POC (FECHA APROX.)	CVE	COMPONENTE	IMPACTO	ESTADO
BlueHammer (7-abr)	CVE-2026-33825	Defender · motor de remediación	LPE → SYSTEM	PARCHEADO abr · in-wild · KEV
RedSun (abr)	CVE-2026-41091	Defender · rollback de cloud files	LPE → SYSTEM	PARCHEADO OOB 21-may · in-wild · KEV
UnDefend (abr)	CVE-2026-45498	Defender · DoS del motor	Evasión (ciega Defender)	PARCHEADO OOB 21-may · in-wild · KEV
YellowKey (13-may)	CVE-2026-45585	BitLocker · WinRE	Bypass físico de cifrado	PARCHEADO jun
GreenPlasma (13-may)	CVE-2026-45586	Framework CTFMON	LPE → SYSTEM	PARCHEADO jun
MiniPlasma (13-may)	CVE-2020-17103	Cloud Files Mini Filter Driver	LPE → SYSTEM	PARCHEADO jun (regresión de 2020)
RoguePlanet (9-jun)	CVE-2026-50656	Defender · pipeline de cuarentena	LPE → SYSTEM	SIN PARCHE · solo PoC
GreatXML (10-jun)	*sin CVE*	BitLocker + WinRE + Defender Offline	Bypass físico de cifrado	SIN PARCHE · TVM-2026-0001

Tres cosas de esta tabla valen más que la tabla misma:

1. **Cinco de las ocho piezas son la misma idea.** BlueHammer, RedSun, RoguePlanet y buena parte de la familia Plasma son variaciones de una condición de carrera TOCTOU en operaciones de fichero privilegiadas de Defender. Parchear un CVE no mata la clase: el actor reencontró el mismo *primitive* por otra ruta de código.
2. **MiniPlasma reutiliza un CVE de 2020.** CVE-2020-17103 es real, del *Cloud Files Mini Filter Driver*, parcheado en noviembre de 2020. Que un PoC de 2026 lo resucite indica un parche incompleto o una regresión silenciosa.
3. **Las dos piezas de BitLocker no son LPE: son bypass físico.** No escalan privilegios; leen el disco cifrado con acceso al arranque, mientras BitLocker sigue reportando «protección activada». Cambian el modelo de amenaza por completo.

5 · El patrón técnico: una sola clase de vulnerabilidad

Conviene entender el mecanismo una sola vez, porque se repite. El objetivo del atacante es siempre el mismo: **hacer que un proceso privilegiado de Windows escriba o ejecute algo bajo su control, con privilegios que el atacante no tiene**. Casi todas las piezas de Defender lo consiguen encadenando tres *primitives* legítimos y bien documentados.

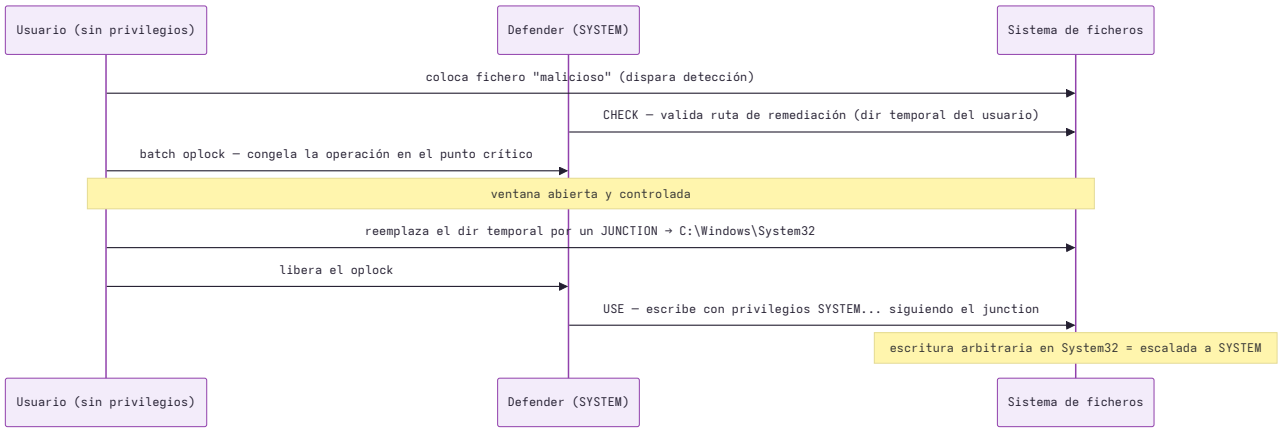
5.1 · TOCTOU, el hueco entre comprobar y usar

TOCTOU (*Time-Of-Check to Time-Of-Use*, **CWE-367**) es una condición de carrera: un programa **comprueba** una condición sobre un recurso (¿esta ruta apunta a mi carpeta temporal?) y **más tarde actúa** sobre él (escribe ahí) **asumiendo que nada cambió en el intervalo**. Si el atacante cambia el recurso entre la comprobación y el uso, el programa actúa sobre algo distinto de lo que validó. En un proceso normal esa ventana es de microsegundos; el «arte» del exploit está en ensancharla y controlarla.

5.2 · Oplocks y junctions: cómo se ensancha la ventana

- **Oplock** (*opportunistic lock*, variante *batch*). Mecanismo legítimo de caché de ficheros de Windows que permite **pausar** una operación cuando otro proceso la toca. El atacante lo usa al revés de su propósito: como un **freno programable** para congelar a Defender exactamente en el instante crítico.
- **Junction** (*junction point* de NTFS). Enlace de directorio (*reparse point*) que **redirige** una ruta a otra. Un usuario sin privilegios puede crearlos en carpetas donde escribe —como su propio `%TEMP%`—.

La receta conceptual: **congelar a Defender con el oplock, cambiar bajo sus pies el directorio real por un junction hacia `System32`, soltar el freno.** Defender reanuda, «sigue» el junction y escribe donde no debía —con sus privilegios, no los del atacante—. Una carrera probabilística se convierte así en una ventana determinista. En secuencia:



5.3 · Windows Error Reporting como salida a SYSTEM

Escribir en `System32` es un medio, no el fin. Varias piezas (MiniPlasma y RoguePlanet, de forma explícita) convierten esa escritura en ejecución como SYSTEM apoyándose en **Windows Error Reporting (WER)**: colocan un `wermgr.exe` suplantado donde WER lo buscará, y disparan la tarea programada `QueueReporting` —que corre como SYSTEM— para que lo ejecute. WER es, en esta campaña, un **vehículo recurrente de ejecución como SYSTEM**. La consecuencia defensiva es directa: **vigilar `wermgr.exe` fuera de `System32` cubre varias piezas a la vez.**

> *Parchear un CVE y creer que se cerró la puerta es contar el último paso e ignorar los mil anteriores. El defensor tapa el agujero que vio; el atacante nunca buscó ese agujero — buscó la clase, y la clase tiene más de una entrada. Mientras el primitive siga siendo «un proceso con privilegios valida una ruta y después escribe sin volver a mirar», contar CVE es contar síntomas.*

6 · Análisis por pieza

6.1 · BlueHammer y RedSun — remediación y rollback de Defender

BlueHammer (`CVE-2026-33825`, CVSS 7.8) y **RedSun** (`CVE-2026-41091`, CVSS 7.8) son dos LPE en Defender que llevan a SYSTEM. Comparten el *primitive* (oplock + junction) pero por rutas de código distintas, y eso explica por qué parchear una no mató a la otra:

	BLUEHAMMER	REDSUN
Componente	Motor de **remediación** de ficheros	Mecanismo de **rollback de cloud files**
Naturaleza (NVD)	Race condition / TOCTOU en la escritura de remediación	*Link-following* (seguimiento de enlace)
Gatillo	Fichero que dispara detección + remediación	Fichero detectado y luego sustituido por un *placeholder* de cloud file; el abuso ocurre en el rollback
Impacto extra	Escritura en System32 + lectura del hive **SAM**	Escritura en System32 → SYSTEM
Parche	Patch Tuesday abril 2026	OOB 21-may-2026

El acceso de lectura al hive **SAM** convierte a BlueHammer en algo más que una LPE: habilita **robo de credenciales** locales (MITRE T1003.002), volcado offline y pivote lateral.

Analizados los PoC públicos en laboratorio, cada pieza deja artefactos concretos y resilientes que alimentan §10:

- **BlueHammer conduce el motor de actualización de Defender.** Trae una actualización legítima de señuelo por COM (`Microsoft.Update.Session`) y hace una **llamada RPC por ALPC** a la interfaz de Defender (`ServerMpUpdateEngineSignature`) que provoca la escritura privilegiada; el freno es un **batch oplock** sobre el fichero de firmas `mpasbase.vdm` y sobre `RstrtMgr.dll`; el gatillo, un EICAR en un fichero `foo.exe`. El SAM se lee **offline con `offreg`** desde una copia de sombra, y la shell se materializa cambiando/restaurando la contraseña del Administrador (`SamiChangePasswordUser`) o con un **servicio efímero de nombre GUID**.
- **RedSun llega por la API de Cloud Files.** Registra un *sync root* falso (`CfRegisterSyncRoot`, *ProviderName* burlón `SERIOUSLYMSFT`), marca el EICAR como *placeholder* de nube y abusa de que Defender lo **reescribe en su ubicación** en vez de ponerlo en cuarentena; con el junction hacia `System32`, esa reescritura cae sobre `TieringEngineService.exe`, que RedSun dispara por **COM local server** para ejecutarlo como SYSTEM.

> **Actualización · 7-jul-2026.** CISA elevó BlueHammer de «explotado in-wild» a **arma confirmada de ransomware**: entró en el catálogo KEV el 22 de abril con plazo de 21 días para las agencias federales, y a finales de junio confirmó campañas de ransomware que lo usan activamente. RedSun, por su parte, ya cuenta con **PoC público en GitHub**, lo que baja aún más la barrera de reutilización.

6.2 · UnDefend — el DoS silencioso

UnDefend (`CVE-2026-45498`) es la pieza que un lector apresurado ignoraría por su puntuación baja. No escala ni ejecuta: es un **denial-of-service contra el propio Defender** que lo deja inservible **sin mostrar ninguna alerta**. El panel dice «protección activada»; la protección, en la práctica, no está. Explota una confianza cómoda: que el estado que un control reporta de sí mismo sea cierto. «El servicio corre» y «el servicio hace su trabajo» son dos afirmaciones distintas, y UnDefend las separa.

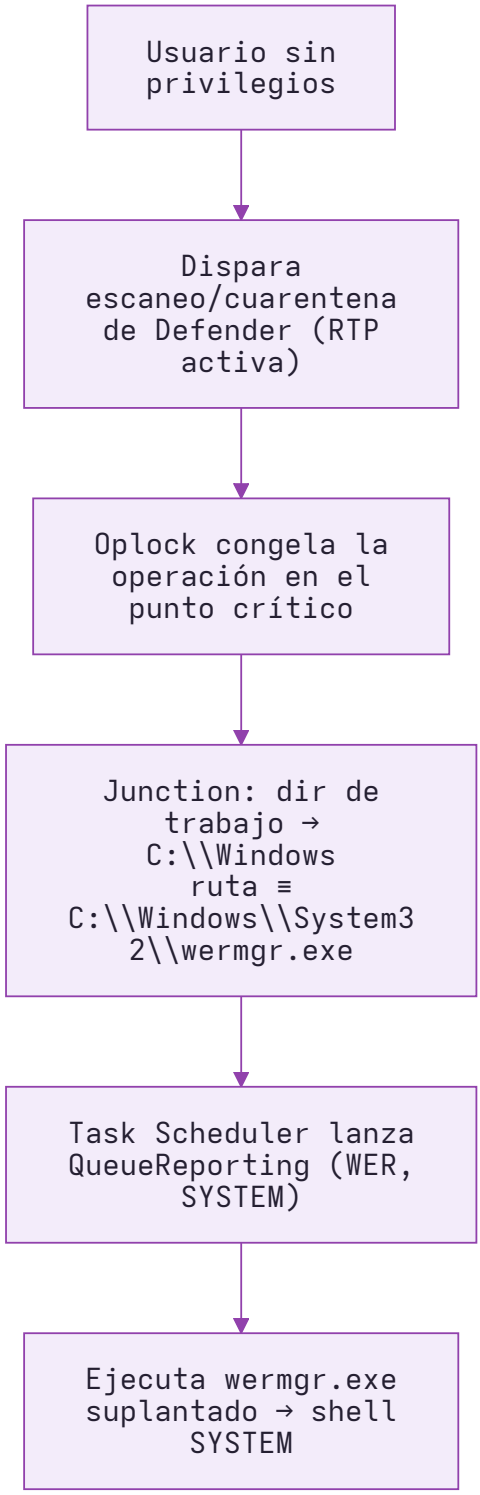
Su valor real no está en la vulnerabilidad aislada, sino en su **lugar en la kill chain**: colocado *antes* de BlueHammer/RedSun, invalida todas las detecciones basadas en firma de los pasos siguientes. Huntress lo vio así: primero escalar, luego cegar con UnDefend, luego tunelizar.

Analizado el PoC en laboratorio, el «cómo» es un abuso de los **bloqueos de rango de bytes** —que Windows respeta de forma absoluta, aun frente a SYSTEM— y deja un rastro observable. En **modo pasivo** vigila el directorio `Definition Updates` con `ReadDirectoryChangesW` y aplica `LockFileEx` / `LOCKFILE_EXCLUSIVE_LOCK` a cada firma nueva; los blancos son `mpavbase.vdm`, los respaldos `\Backup\mpavbase.lkg` / `.vdm` y el directorio `System32\MRT`. En **modo agresivo** registra `NotifyServiceStatusChangeW(SERVICE_NOTIFY_STOPPED)` sobre `WinDefend` para re-bloquear el motor cuando una actualización de plataforma lo detiene, impidiendo que reanude. El README del autor añade un dato con peso defensivo: afirma —sin publicarlo— una técnica para **falsear el estado en la consola EDR**; verificarlo es imposible, pero refuerza la tesis de recoger la salud de Defender de forma centralizada y no fiarse del semáforo local.

> **Actualización · 7-jul-2026.** La NVD reevaluó el CVE al alza: figura ahora como **CVSS 7.5 (alto)**, tipo «consumo no controlado de recursos» (CWE-400), no el 4.0 con que varios proveedores lo publicaron al inicio. El reajuste, si acaso, refuerza el argumento: incluso el puntaje «bajo» era engañoso. CISA le fijó plazo de parcheo del 3 de junio; la plataforma corregida es la `4.18.26040.7`.

6.3 · RoguePlanet — la carrera que sobrevive al parche

RoguePlanet (`CVE-2026-50656`) es la cuarta pieza contra Defender y la que mejor confirma la tesis: el mismo *primitive* por una puerta nueva, el **pipeline de cuarentena**. La vulnerabilidad vive en el hueco entre cuando Defender crea un artefacto de cuarentena y cuando valida dónde aterrizó. El exploit construye una estructura de directorios que imita `System32`, convierte el directorio de trabajo en un junction hacia `C:\Windows` —de modo que la ruta se vuelve transparentemente idéntica a `C:\Windows\System32\wermgr.exe`— y usa el Task Scheduler para lanzar la tarea `QueueReporting` de WER, que ejecuta el `wermgr.exe` suplantado como SYSTEM.



Consecuencia práctica importante: un *application allowlisting* basado en **ruta** puede ser burlado por esa suplantación. WDAC/AppLocker ayuda, pero **con reglas de publisher/hash, no de ruta**. Y como no hay parche, la **detección comportamental es el único control real**.

Analizado el PoC público en laboratorio —a nivel de qué observa un defensor, no de cómo se arma—, la cadena deja una huella más resiliente que el named pipe o los nombres de fichero. Cuatro artefactos de comportamiento la anclan, y alimentan las reglas de §10:

- **Conduce al propio motor de Defender.** Para forzar la cuarentena que luego secuestra, el PoC carga `MpClient.dll` desde el directorio de Defender y encadena su API de gestión (`MpManagerOpen` → `MpScanStart` → `MpScanResult` → `MpThreatOpen` / `MpThreatEnumerate` → `MpCleanOpen` / `MpCleanStart`). Un proceso ajeno a Defender que carga esa DLL y dispara un *scan+clean* es una anomalía casi nunca benigna.
- **Gatillo por EICAR disfrazado.** El fichero que se hace detectar es un **EICAR** escrito con el nombre `wermgr.exe` bajo un directorio temporal del usuario —una combinación (EICAR + nombre `wermgr.exe` + `%TEMP%`) que no ocurre por accidente.
- **Montaje de ISO desde `%TEMP%`.** Escribe un `.iso` embebido en `%TEMP%\RP_<GUID>` y lo adjunta con VirtDisk sin letra de unidad. El README del autor confirma la condición sin querer: el PoC *no funciona en Windows Server porque un usuario estándar no puede montar una ISO ahí*.
- **La tarea WER se lanza a mano.** El salto a SYSTEM es un `Run()` **on-demand** de `\Microsoft\Windows\Windows Error Reporting\QueueReporting` desde un token de usuario interactivo —una tarea que normalmente dispara el sistema—, seguido de un `wermgr.exe` SYSTEM. Artefacto estable y de alta fidelidad.

> **Actualización · 7-jul-2026.Sigue sin parche.** Microsoft mantiene el fix «en desarrollo» sin fecha pública; se espera en el Patch Tuesday del 14 de julio o en un OOB antes (los precedentes de esta campaña —RedSun, UnDefend— fueron OOB). Continúa **sin explotación in-the-wild** confirmada por Microsoft y validado contra Windows 10/11 totalmente parcheados. Fue, además, la pieza que «drenó» al actor y lo llevó a cancelar la divulgación masiva del 14 de julio.

6.4 · GreenPlasma y MiniPlasma — CTFMON y el CVE zombie de 2020

MiniPlasma (`CVE-2020-17103`) es la mejor historia de la campaña: reutiliza **sin cambios** un PoC de Project Zero de 2020 contra el *Cloud Files Mini Filter Driver* (`cldflt.sys`), «parcheado» en noviembre de 2020. Seis años después seguía elevando a SYSTEM. Según GuardSix, el PoC original «sigue funcionando sin cambios, lo que significa que el fix o nunca se aplicó correctamente o fue silenciosamente revertido». El mecanismo: ganando una carrera al alternar tokens de impersonación durante `CfAbortOperation()`, el driver escribe en `HKEY_USERS\.DEFAULT`; eso se convierte en envenenamiento de variables de entorno (redirigir `%windir%`) que, otra vez, hace que **WER ejecute un `wermgr.exe` malicioso como SYSTEM.**

GreenPlasma (`CVE-2026-45586`) abusa de `ctfmon.exe`, el servicio de entrada de texto que corre como SYSTEM. Crea un enlace simbólico del Object Manager hacia `CTF.AsmListCache.FMPWinlogon` para que `ctfmon` abra, sin saberlo, un objeto controlado por el atacante, y usa un symlink de registro para saltarse los DACL del hive de *Policies*. Su PoC es menos «llave en mano» —todavía dispara un prompt de UAC—, pero demuestra que un proceso SYSTEM que abre objetos por nombre sin validar su origen es manipulable.

Analizado el PoC en laboratorio (es **parcial** a propósito: el autor quitó el código del shell SYSTEM completo y dejó la primitiva de creación arbitraria de secciones), los artefactos concretos son de registro y espacio de nombres: resetea la DACL de `HKCU\Software\Policies\Microsoft\CloudFiles` y recrea `…\CloudFiles\BlockedApps` como enlace simbólico de registro (`REG_OPTION_CREATE_LINK`, valor

`SymbolicLinkValue` / `REG_LINK`) para escribir **`DisableLockWorkstation=1`** en `Policies\System`; y crea el enlace del Object Manager `\Sessions\<id>\BaseNamedObjects\CTF.AsmListCache.FMPWinlogon<id>` (target por defecto `CTFMON_DEAD`). La shell sale por `ShellExecuteEx` verbo `runas` sobre `conhost.exe`. La señal defensiva resiliente es la de registro —symlink (`SymbolicLinkValue`) o escritura de `DisableLockWorkstation` por un proceso de usuario— más que los nombres del Object Manager (frágiles).

Lección de ingeniería. Un fix no está cerrado hasta que se **verifica** que el PoC original ya no funciona —idealmente con un test de regresión en cada build—. MiniPlasma es el caso de libro de por qué «marcado como parcheado» no es «parcheado»: el CVE estuvo seis años en estado zombie. Aplica igual a tu propio código.

Actualización · 7-jul-2026. Nada urgente aquí, y es buena noticia dentro de la campaña: ambas siguen **sin explotación in-the-wild** (se quedaron en PoC) y salieron dentro de un Patch Tuesday de junio grande (~200 CVE, 6 zero-days). Microsoft caracteriza oficialmente GreenPlasma como una vulnerabilidad de *link-following* en el Collaborative Translation Framework (CTFMON). La regresión de `CVE-2020-17103` quedó confirmada por varios análisis independientes.

6.5 · YellowKey y GreatXML — bypass de BitLocker vía WinRE

Estas dos no escalan privilegios: **leen el disco cifrado** mientras BitLocker reporta «protección activada». Ninguna rompe la criptografía; ambas atacan el **Windows Recovery Environment (WinRE)**, código de confianza que corre en el arranque temprano, antes de que se establezcan las fronteras de seguridad normales. Pero **no son intercambiables**:

	YELLOWKEY (`CVE-2026-45585`)	GREATXML (SIN CVE)
Requisito	**Acceso físico** breve	**Admin una vez** en el equipo
Config vulnerable	BitLocker **TPM-only** (sin PIN)	Cualquiera donde se corrió Defender Offline Scan
Naturaleza	Bypass de cifrado *in situ*	**Backdoor de persistencia** que sobrevive a IR
Estado	Parcheado (jun)	Sin parche; validación externa incompleta

YellowKey abusa de que WinRE confía en `autofstx.exe` para procesar logs de NTFS transaccional (TxF) desde unidades conectadas. El atacante planta logs manipulados en un USB o la partición EFI; al reproducirlos, `autofstx.exe` borra `winpeshl.ini` y WinRE cae a un símbolo del sistema sin restricciones, con el volumen BitLocker **ya descifrado por el TPM**. **GreatXML** planta un `unattend.xml` y un `ReAgent.xml` manipulados en la partición de recuperación; al reiniciar en WinRE se procesan antes de cualquier logon y lanzan un shell SYSTEM sobre el volumen cifrado. Lo peligroso de GreatXML no es el acceso puntual, es la **persistencia**: sobrevive a la rotación de credenciales y a la pérdida del acceso remoto.

Analizados los artefactos reales en laboratorio, ambos dejan huellas concretas y estables. El **`unattend.xml` de GreatXML** es un autounattend generado con una herramienta pública que, por *passes*: escribe `X:\pe.cmd` (`start …\conhost.exe`) para abrir una consola en el WinPE ya desbloqueado; crea las cuentas locales **`Admin`** / **`User`**

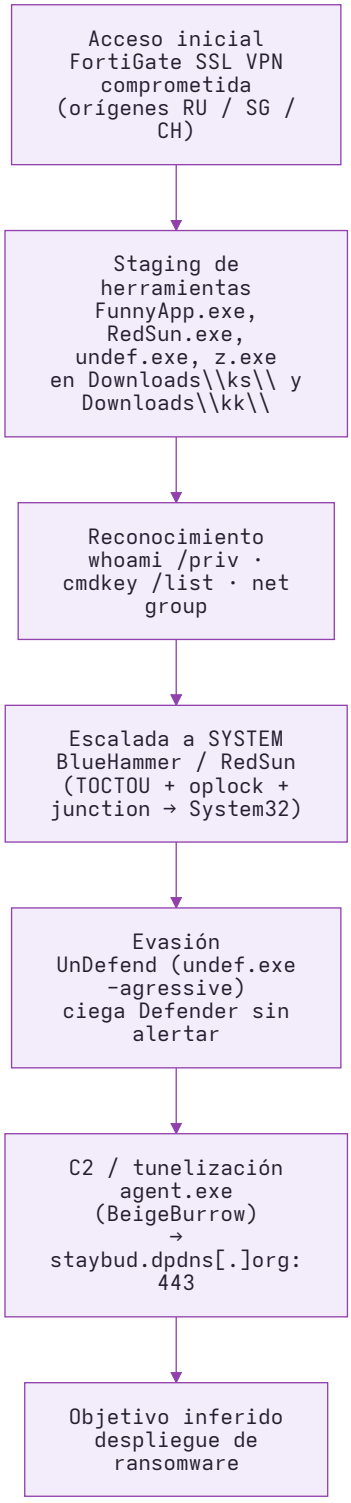
con contraseña en blanco y **AutoLogon de** `Admin` (la persistencia que sobrevive a IR); y con `FirstLogon.ps1` **se autoborra** (`AutoLogonCount=0` y borrado de `C:\Windows\Panther\unattend.xml` / `unattend-original.xml`). El residuo forense *post-boot* —cuentas sin contraseña, `AutoAdminLogon`, scripts en `C:\Windows\Setup\Scripts\` — es lo cazable. En **YellowKey**, el artefacto es un árbol CLFS/TxF (`System Volume Information\FsTx\<GUID>\FsTxLogs\` con `FsTxLog.blf` / `FsTxKtmLog.blf` y contenedores); la anomalía no es su contenido sino su **ubicación** en medios extraíbles o la partición EFI/recuperación, no en la raíz de un volumen NTFS fijo.

Actualización · 7-jul-2026.YellowKey: quedó con **CVSS 6.8** y afecta a Windows 11 24H2/25H2/26H1 y Windows Server 2025. Antes del parche de junio, Microsoft publicó (20-may) una mitigación provisional —un script que reedita el registro SYSTEM offline de la imagen de WinRE— y afirma que pasar a **TPM+PIN bloquea eficazmente** el ataque. El shell irrestricto se dispara manteniendo **CTRL** durante el arranque de WinRE. **GreatXML:** sigue sin CVE y sin parche; Microsoft dice «investigar la validez» del reporte y lo rastrea con el id interno `TVM-2026-0001`, aún sin mapear a un CVE público.

El modelo de amenaza que rompe esta pareja. «Cifrado en reposo» protege contra un atacante que se lleva el disco, no contra uno que controla el *arranque* de la máquina. WinRE es código de confianza que corre antes que tus defensas; cualquier cosa que confíe en datos externos en esa fase (logs TxF, `unattend.xml`) es superficie pre-boot. **BitLocker sin PIN + acceso al arranque ≈ sin cifrado.**

7 · La cadena real: la intrusión documentada por Huntress

La teoría es una cosa; lo que Huntress documentó en una intrusión real de mediados de abril es otra, y es lo que importa para cazar. El acceso inicial **no** fue ninguno de estos zero-days: fue una **cuenta de FortiGate SSL VPN comprometida**, usada desde tres orígenes geográficos en ventanas cortas. Los zero-days entraron *después*, para escalar y cegar la defensa.



Fíjese en el orden: **primero escalan, luego ciegan, luego tunelizan**. UnDefend es el eslabón silencioso —un DoS que apaga la luz un rato— y por eso el de mayor valor operativo pese a su CVSS. Un número que ordena por severidad terminó ordenando por atención: lo que puntúa bajo se mira tarde.

8 · Indicadores de compromiso consolidados

Los siguientes indicadores provienen de la intrusión real documentada por Huntress. **Valídelos contra su entorno antes de bloquear nada**; los nombres de binario son triviales de cambiar y deben usarse como pista, no como muralla.

Ficheros y rutas de staging

```
# staging (nombres cosméticos — renombrables por el atacante)
%USERPROFILE%\Pictures\FunnyApp.exe          (variante BlueHammer)
%USERPROFILE%\Downloads\RedSun.exe
%USERPROFILE%\Downloads\ks\undef.exe          (UnDefend; flags -h / -agressive)
%USERPROFILE%\Downloads\kk\undef.exe
%USERPROFILE%\Downloads\ks\z.exe
agent.exe                                     (tunelizador BeigeBurrow)
```

Red / C2

```
staybud.dpdns[.]org:443     C2 (agent.exe -server ... -hide)
78.29.48[.]29               VPN origen (Rusia)
212.232.23[.]69             VPN origen (Singapur)
179.43.140[.]214            VPN origen (Suiza)
```

Otros artefactos

```
SHA-256  a2b6c7a9c4490df70de3cdbfa5fc801a3e1cf6a872749259487e354de2876b7c
Firma    Exploit:Win32/DfndrPEBluHmr.BZ     (detección de Defender para BlueHammer)
Vector   Credenciales de FortiGate SSL VPN comprometidas (acceso inicial)
```

No confunda artefacto de PoC con IoC de intrusión. El *named pipe* `\\.\pipe\RoguePlanet` que circula en algunos análisis es un artefacto del PoC público de RoguePlanet, **no** un indicador visto en un ataque real. Bloquear por ese nombre es cosmético: cualquier variante lo renombra. La detección resiliente es comportamental (un shell hijo de `MsMpEng.exe`), no el nombre del pipe.

9 · Mapa MITRE ATT&CK de la campaña

TÁCTICA	TÉCNICA	DÓNDE APARECE
Initial Access	T1190 — Exploit Public-Facing Application	FortiGate SSL VPN
Initial Access	T1078 — Valid Accounts	Credenciales VPN robadas
Execution / Ingress	T1105 — Ingress Tool Transfer	FunnyApp.exe, RedSun.exe, undef.exe, agent.exe
Privilege Escalation	T1068 — Exploitation for Privilege Escalation	BlueHammer, RedSun, GreenPlasma, MiniPlasma, RoguePlanet
Privilege Escalation	T1574 — Hijack Execution Flow	RedSun (`TieringEngineService.exe` reescrito)
Privilege Escalation	T1548 — Abuse Elevation Control Mechanism	GreenPlasma (DACL + symlink de registro, `runas`)
Defense Evasion	T1112 — Modify Registry	GreenPlasma (`DisableLockWorkstation` , `SymbolicLinkValue`)
Privilege Escalation	T1053.005 — Scheduled Task/Job	QueueReporting de WER (MiniPlasma, RoguePlanet)
Execution	T1569.002 — System Services: Service Execution	BlueHammer (servicio efímero GUID)
Defense Evasion	T1562.001 — Impair Defenses: Disable or Modify Tools	UnDefend (bloqueo de firmas/motor)
Impact	T1489 — Service Stop	UnDefend modo agresivo (impide que WinDefend reanude)
Defense Evasion	T1036 — Masquerading	`wermgr.exe` suplantado vía junction
Defense Evasion	T1553.005 — Subvert Trust Controls: contenedor montable (ISO)	RoguePlanet (ISO desde `%TEMP%`)
Defense Evasion	T1564.004 — Hide Artifacts: NTFS ADS	RoguePlanet (`:WDFOO` sobre el señuelo)
Credential Access	T1003.002 — OS Credential Dumping: SAM	BlueHammer (lectura de SAM)
Credential Access	T1555 — Credentials from Password Stores	`cmdkey /list`
Discovery	T1087 — Account Discovery	`whoami /priv` , `net group`
Command and Control	T1090 — Proxy / tunneling	agent.exe (BeigeBurrow) → C2
Impact	T1486 — Data Encrypted for Impact (inferido)	despliegue de ransomware
Collection / pre-OS	T1006 — Direct Volume Access · T1542 — Pre-OS Boot	YellowKey, GreatXML (BitLocker/WinRE)
Persistence	T1136.001 — Create Account: Local Account	GreatXML (`Admin` / `User` sin contraseña, AutoLogon)

Defense Evasion	T1070.004 — Indicator Removal: File Deletion	GreatXML (autoborrado del `unattend.xml`)

10 · Ingeniería de detección

La regla de oro de esta campaña: **caza la clase, no el indicador**. Las reglas comportamentales que siguen son resilientes a que el actor renombre binarios y pipes, y cubren varias piezas a la vez. Se ordenan de más a menos robustas.

10.1 · Shell hijo de MsMpEng.exe (alta confianza)

`MsMpEng.exe` (el servicio de Defender) no lanza shells interactivos. Un `cmd` / `powershell` como hijo directo es un artefacto potente de explotación de esta familia (BlueHammer, RedSun, RoguePlanet) y no depende de ningún IoC.

```
# Sigma (Sysmon EventID 1)
detection:
  selection:
    ParentImage|endswith: '\MsMpEng.exe'
    Image|endswith:
      - '\cmd.exe'
      - '\powershell.exe'
      - '\pwsh.exe'
      - '\wscript.exe'
      - '\cscript.exe'
      - '\mshta.exe'
  condition: selection
level: critical
```

```
// KQL (Defender XDR / Sentinel)
DeviceProcessEvents
| where Timestamp > ago(24h)
| where InitiatingProcessFileName =~ "MsMpEng.exe"
| where FileName in~ ("cmd.exe","powershell.exe","pwsh.exe",
                      "wscript.exe","cscript.exe","mshta.exe")
| project Timestamp, DeviceName, FileName, ProcessCommandLine,
          AccountName, InitiatingProcessFileName
```

10.2 · wermgr.exe fuera de ruta (cubre MiniPlasma y RoguePlanet)

`wermgr.exe` legítimo vive en `System32` / `SysWOW64`. Cázalo cuando se escribe o ejecuta desde cualquier otra ruta: es el corazón del truco de suplantación vía WER.

```
// KQL
DeviceProcessEvents
| where Timestamp > ago(30d)
| where FileName =~ "wermgr.exe"
| where not(FolderPath has_any (@"\Windows\System32\", @"\Windows\SysWOW64\"))
| project Timestamp, DeviceName, FolderPath, ProcessCommandLine,
          InitiatingProcessFileName, AccountName
```

10.3 · Los dos pasos que RoguePlanet no puede renombrar (alta fidelidad)

Del análisis del PoC real salen dos señales comportamentales que no dependen de ningún nombre y cubren el flujo sin parche de extremo a extremo. **(a)** El PoC conduce al motor de Defender cargando `MpClient.dll` para forzar la cuarentena que secuestra; fuera de los binarios de Defender casi nada carga esa DLL. **(b)** El salto a SYSTEM es un arranque *on-demand* de la tarea `QueueReporting` de WER por un token de usuario —normalmente la dispara el sistema—.

```
# Sigma — MpClient.dll cargado por proceso ajeno a Defender
title: RoguePlanet — MpClient.dll cargado por proceso no-Defender
id: ne-rp-002-mpclient-load
status: experimental
description: >
  Proceso ajeno a Defender que carga MpClient.dll para conducir el motor
  antimalware (MpScanStart/MpCleanStart). Fuerza la cuarentena que RoguePlanet
  (CVE-2026-50656) secuestra. Comportamental: no depende de nombres del PoC.
references:
  - https://www.cyderes.com/howler-cell/rogueplanet-windows-zero-day
author: Serie Nightmare Eclipse — detección defensiva
date: 2026-07-08
tags:
  - attack.privilege_escalation
  - attack.t1068
logsource:
  product: windows
  category: image_load
detection:
  selection:
    ImageLoaded|endswith: '\MpClient.dll'
  filter_defender:
    Image|endswith: ['\MsMpEng.exe', '\MpCmdRun.exe', '\NisSrv.exe', '\MpDefenderCoreService.exe']
  condition: selection and not filter_defender
falsepositives:
  - Herramientas de administración/EDR que instrumenten Defender. Validar baseline.
level: high
```

```
// KQL — QueueReporting lanzada on-demand por un usuario (no SYSTEM)
DeviceEvents
| where Timestamp > ago(24h)
| where ActionType in ("ScheduledTaskExecuted","ScheduledTaskStarted")
| where AdditionalFields has "QueueReporting"
| where InitiatingProcessAccountName !in ("system","local service","network service")
| project Timestamp, DeviceName, InitiatingProcessFileName,
          InitiatingProcessAccountName, AdditionalFields
```

En `Microsoft-Windows-TaskScheduler/Operational`, la señal equivalente es un EventID 119/100/200 para `\Microsoft\Windows\Windows Error Reporting\QueueReporting` cuyo contexto no sea `SYSTEM`. Como señal complementaria de media-alta confianza: un `.iso` **escrito en el perfil de un usuario y montado** por un proceso no-sistema (canal `Microsoft-Windows-VHDMP-Operational`) precede a la carrera —el PoC monta una ISO para obtener una copia limpia de `wermgr.exe`—.

10.4 · Salud de Defender centralizada (detecta UnDefend)

UnDefend se detecta por **ausencia y por fallo**, no por presencia. No confíes en el semáforo local: recoge el estado de Defender de forma centralizada y alerta sobre firmas que dejan de actualizarse o protección en tiempo real que cae. Regla de negocio simple y potente: **si `AntivirusSignatureLastUpdated` tiene más de 24–48 h en un host con red, es un incidente hasta que se demuestre lo contrario.**

```
# PowerShell — recolección de salud para alertar centralmente
Get-MpComputerStatus | Select-Object `
    AMRunningMode, RealTimeProtectionEnabled, AntivirusEnabled, `
    AMServiceEnabled, AntivirusSignatureLastUpdated, `
    AMProductVersion, IsTamperProtected
```

```
// KQL — fallos de actualización / RTP caída correlacionados
DeviceEvents
| where Timestamp > ago(24h)
| where ActionType in ("AntivirusDefinitionUpdateFailed",
                       "AntivirusRealTimeProtectionDisabled",
                       "AntivirusScanFailed")
| summarize Eventos = count(), Tipos = make_set(ActionType)
    by DeviceName, bin(Timestamp, 1h)
| where Eventos > 2
```

La telemetría de salud caza el *efecto*; el código de UnDefend da además una señal en el *acto*: un proceso **no-`MsMpEng.exe`** manteniendo handles de larga duración con **bloqueo exclusivo** sobre ficheros en `Definition Updates\`, `\Backup\mpavbase.lkg` / `.vdm` o `System32\MRT\`. Con EDR que exponga *file handles*, es un artefacto de alta fidelidad; en modo agresivo, complementa el registro de `NotifyServiceStatusChange` sobre `WinDefend` desde un proceso no administrativo.

10.5 · Escritura inesperada en System32 por Defender (y el sync root de RedSun)

El desenlace del *primitive* es siempre observable: `MsMpEng.exe` crea o reescribe un ejecutable en `System32`. En el PoC de RedSun el blanco concreto es `TieringEngineService.exe` (que luego se dispara por COM para correr como SYSTEM); en la familia WER, `wermgr.exe`. La regla no necesita el nombre:

```
// KQL — Defender no debería crear/reescribir binarios en System32
DeviceFileEvents
| where Timestamp > ago(24h)
| where FolderPath startswith @"C:\Windows\System32\"
| where InitiatingProcessFileName =~ "MsMpEng.exe"
| where ActionType in ("FileCreated","FileModified")
| where FileName endswith ".exe" or FileName endswith ".dll"
```

Como señal previa específica de RedSun: el registro de un **sync root de Cloud Files** (`CfRegisterSyncRoot`, carga de `CldApi.dll`) por un proceso que **no es un proveedor de nube conocido** —en el PoC, bajo el *ProviderName* burlón `SERIOUSLYMSFT`— precede al abuso del *rollback*. Fuera de OneDrive y clientes equivalentes, casi nada registra un sync root.

10.6 · Reglas YARA para los artefactos del PoC público

Las reglas Sigma/KQL de arriba cazan **comportamiento**; las YARA que siguen cazan **ficheros**, y por eso son deliberadamente el eslabón más frágil de este arsenal. Dos aclaraciones antes de copiarlas: **(1)** no son detecciones novedosas —la idea de cazar un binario de sistema suplantado o los artefactos de WER es un género público bien establecido; aquí solo se **adapta ese patrón conocido** al vocabulario de la campaña (véase, como prior art, la familia `APT_*_ForensicArtefacts_WER` de la signature-base de Nextron / Florian Roth)—; **(2)** las cadenas específicas se **recopilan de los artefactos del PoC público y del reporting de Huntress** citados en este informe (§8), no de telemetría propia. Sirven para triaje de muestras en reposo (una carpeta de descargas, un adjunto, un artefacto de sandbox), **no como control primario**: el actor renombra binarios y pipes en minutos, así que un match confirma «esto es el PoC público tal cual» y un no-match no dice nada. El campo `author` refleja quién las **recopiló/adaptó** para este documento, no la autoría de la técnica.

```
import "pe"
import "hash"

/* RECOPILACIÓN, no autoría: cadenas de los PoC públicos de Nightmare Eclipse y
   del tooling que Huntress observó en la intrusión real (nombres de binario,
   flags, C2, pipe) — todas provienen del reporting citado en §8/§15.
   ENCUADRE: firma FRÁGIL, de conveniencia. El pipe RoguePlanet es artefacto de
   PoC, NO IoC de intrusión (ver §8). Úsese como complemento del comportamiento. */
rule NightmareEclipse_PoC_Tooling
{
    meta:
        author       = "dust115 — recopilación (fennek.org)"
        description  = "Cadenas de los PoC públicos (BlueHammer/RedSun/UnDefend/RoguePlanet) y del tooling
visto por Huntress"
        source       = "artefactos del PoC público + reporting de Huntress (ver §8/§15)"
        tlp          = "CLEAR"
        confidence   = "low — IoC frágil, renombrable; complementario"
    strings:
        $pipe  = "\\\\.\\pipe\\RoguePlanet" ascii wide      // artefacto de PoC, NO IoC de intrusión real
        $rp    = "RoguePlanet" ascii wide nocase
        $undef = "-agressive" ascii wide                    // flag del PoC UnDefend (sic)
        $tun   = "BeigeBurrow" ascii wide nocase            // tunelizador (agent.exe)
        $c2    = "staybud.dpdns" ascii wide nocase          // C2 reportado por Huntress
        $wer   = "QueueReporting" ascii wide                // tarea WER abusada (contexto)
        $cf    = "CfAbortOperation" ascii wide              // MiniPlasma / cldflt.sys
        $rsprov = "SERIOUSLYMSFT" ascii wide                // ProviderName del sync root de RedSun
(verificado en código)
        $rspipe = "\\??\\pipe\\REDSUN" ascii wide           // named pipe de RedSun (verificado en código)
    condition:
        2 of ($pipe, $rp, $undef, $tun, $c2, $rsprov, $rspipe)
        or (1 of ($pipe, $rp, $undef, $tun, $c2, $rsprov, $rspipe) and 1 of ($wer, $cf))
}

/* RECOPILACIÓN, no autoría: cadenas VERIFICADAS contra el ejecutable del PoC
   público de RoguePlanet (analizado en laboratorio, no nombres reportados).
   Incluye un artefacto forense involuntario: la ruta del PDB delató el nombre
   interno del proyecto, "ScanMan". Firma FRÁGIL, de conveniencia. */
rule NightmareEclipse_RoguePlanet_PoC
{
    meta:
        author       = "dust115 — recopilación (fennek.org)"
        description  = "Cadenas del ejecutable del PoC público RoguePlanet (CVE-2026-50656)"
        source       = "artefactos del binario del PoC público analizado en laboratorio"
        tlp          = "CLEAR"
        confidence   = "low — IoC frágil, renombrable; complementario"
    strings:
        $pdb   = "ScanMan\\x64\\Release\\RoguePlanet.pdb" ascii
        $pipe  = "\\\\.\\pipe\\RoguePlanet" wide            // artefacto de PoC, NO IoC de intrusión real
        $ads   = ":WDFOO" wide                              // ADS sobre el wermgr señuelo
        $tmp   = "\\??\\%TEMP%\\RP_" wide                   // prefijo del dir/ISO temporal
        $wd    = "wdtest_temp" wide
        $ok    = "Exploit succeeded." ascii
    condition:
        uint16(0) == 0x5A4D and 2 of them
}

/* ADAPTACIÓN de un patrón público, no técnica propia: wermgr.exe es el vehículo
```

```
legítimo vive FIRMADO en System32/SysWOW64; un wermgr.exe SIN firma
   Authenticode es un candidato de suplantación. La idea (binario de sistema
   suplantado / artefactos WER) es prior art conocido; aquí solo se instancia.
   ENCUADRE: regla de CAZA, no de bloqueo — triáguese ruta y firma antes de
   actuar (hay wermgr sin firmar legítimos en imágenes muy viejas). */
rule Suspicious_Unsigned_Wermgr
{
    meta:
        author      = "dust115 — adaptación (fennek.org)"
        description = "wermgr.exe sin firma Authenticode — posible suplantación WER (T1036)"
        credit      = "patrón público: cf. signature-base APT_HAFNIUM_ForensicArtefacts_WER (Nextron/F.
Roth)"
        tlp         = "CLEAR"
        confidence  = "medium — requiere triaje de ruta y firma"
    condition:
        pe.is_pe
        and (pe.version_info["OriginalFilename"] == "wermgr.exe"
             or pe.version_info["InternalName"] == "wermgr.exe")
        and pe.number_of_signatures == 0
}

/* La firma MÁS frágil posible: el hash exacto de la muestra de BlueHammer
   reportada por Huntress. Un solo byte distinto la evade; se incluye como
   ejemplo del techo de utilidad de un IoC de hash, no como defensa. */
rule NightmareEclipse_BlueHammer_KnownSample
{
    meta:
        author      = "dust115 — recopilación (fennek.org)"
        description = "Muestra concreta de BlueHammer (Huntress) — IoC de hash, un único fichero"
        source      = "hash publicado por Huntress (ver §8)"
        tlp         = "CLEAR"
        confidence  = "informational — un solo binario; se evade recompilando"
    condition:
        hash.sha256(0, filesize) ==
            "a2b6c7a9c4490df70de3cdbfa5fc801a3e1cf6a872749259487e354de2876b7c"
}
```

10.7 · Integridad de la partición de recuperación (YellowKey / GreatXML)

Estos ataques ocurren **antes** del arranque completo de Windows: el EDR no está corriendo, no hay proceso que observar. El enfoque cambia de «detectar» a «prevenir e integridad»: monitoriza escrituras a `\Recovery\` y a la partición de recuperación (especialmente `unattend.xml` / `ReAgent.xml`), audita qué endpoints han ejecutado Defender Offline Scan (superficie de GreatXML) y alerta sobre montaje de USB y accesos a la partición EFI en equipos sensibles.

Como el `unattend.xml` de GreatXML **se autoborra**, la caza práctica es el residuo *post-boot*: cuentas locales con contraseña en blanco, AutoLogon y los scripts de setup que deja atrás.

```
# Residuo forense de GreatXML tras el arranque
Get-LocalUser | Where-Object { -not $_.PasswordRequired } | Select-Object Name, Enabled
Get-ItemProperty 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon' |
    Select-Object AutoAdminLogon, DefaultUserName, AutoLogonCount
Test-Path 'C:\Windows\Panther\unattend.xml',
          'C:\Windows\Setup\Scripts\Specialize.ps1',
          'C:\Windows\Setup\Scripts\FirstLogon.ps1'
```

Para YellowKey, la señal es de **ubicación**: un árbol `System Volume Information\FsTx\<GUID>\FsTxLogs\` (CLFS `.blf` + contenedores `FsTx*Container*`) en medios extraíbles o en la partición EFI/recuperación —donde esos logs de transacción NTFS no pertenecen—.

> **Sobre los IoC frágiles.** Detectar el *named pipe* `RoguePlanet` o los nombres `undef.exe` / `FunnyApp.exe` sirve para cazar el PoC *tal cual*, jamás como control primario. La sombra que un atacante no puede renombrar es el comportamiento: un hijo de `MsMpEng.exe`, un `wermgr.exe` donde no vive, un Defender que dejó de actualizar.

11 · Mitigación y plan de acción

11.1 · Acciones inmediatas

- **Parchea la plataforma de Defender.** Verifica `4.18.26040.7` o superior —cubre BlueHammer, RedSun y UnDefend—: `(Get-MpComputerStatus).AMProductVersion`.
- **Aplica los Patch Tuesday de abril y junio 2026** (BlueHammer; YellowKey/GreenPlasma/MiniPlasma) y el **OOB del 21-may** (RedSun/UnDefend).
- **Activa y monitoriza Tamper Protection.** Dificulta —no elimina— la manipulación de Defender.

11.2 · Para lo que sigue sin parche (RoguePlanet, GreatXML)

AMENAZA	CONTROL	NOTA
RoguePlanet	Detección comportamental (§10.1, §10.2)	Es el control real mientras no hay parche
RoguePlanet	WDAC/AppLocker por **publisher/hash**	Las reglas por ruta las burla el junction
GreatXML	Restringir admin local	Sin admin no se planta el backdoor
GreatXML	Auditar/limpiar la partición de recuperación	Verifica integridad tras cualquier acceso admin sospechoso
YellowKey	BitLocker **TPM+PIN** + password UEFI + no boot USB	Corta el vector físico y el escenario TPM-only
Ambas BitLocker	Restringir acceso físico; endurecer/deshabilitar WinRE donde no sea necesario	`reagentc /disable` en entornos de alta seguridad

11.3 · Principio transversal

Caza comportamental, no solo IoC. Alerta sobre shells hijos de `MsMpEng.exe`, sobre `wermgr.exe` fuera de ruta y sobre fallos inexplicados de actualización de Defender correlacionados con otros eventos. Son resilientes a que el actor renombre binarios y pipes, y cubren tanto lo parcheado como lo que vendrá por la misma clase.

12 · Estado a julio de 2026

PIEZA	CVE	PARCHE	IN-WILD	NOTAS AL 7-JUL
BlueHammer	`CVE-2026-33825`	Sí (abr)	Sí · ransomware	KEV 22-abr; confirmado como arma de ransomware por CISA
RedSun	`CVE-2026-41091`	Sí (OOB 21-may)	Sí	PoC público en GitHub
UnDefend	`CVE-2026-45498`	Sí (OOB 21-may)	Sí	NVD lo reevaluó a CVSS 7.5 (CWE-400)
YellowKey	`CVE-2026-45585`	Sí (jun)	No	CVSS 6.8; mitigación provisional 20-may; TPM+PIN lo bloquea
GreenPlasma	`CVE-2026-45586`	Sí (jun)	No	link-following en CTFMON
MiniPlasma	`CVE-2020-17103`	Sí (jun)	No	Regresión de un CVE de 2020 confirmada
RoguePlanet	`CVE-2026-50656`	**No**	No	Fix en desarrollo; esperado Patch Tue jul o OOB
GreatXML	*sin CVE*	**No**	No	MS «investiga la validez»; id interno `TVM-2026-0001`

13 · Conclusiones: cuatro lecciones

1. **Caza clases de vulnerabilidad, no CVE.** Cinco de las ocho piezas son el mismo *primitive* (TOCTOU + oplock + junction, con WER como salida a SYSTEM). Una detección comportamental sólida —shell hijo de `MsMpEng.exe`, `wermgr.exe` fuera de ruta— cubre lo que cinco firmas de CVE no.
2. **El estado autorreportado miente.** UnDefend y las piezas de BitLocker demuestran que «protección activada» y «protegido» son cosas distintas. La verdad está en la telemetría de segundo orden: ¿sigue actualizando?, ¿sigue emitiendo los eventos que debería?, ¿hay huecos?
3. **Un parche sin verificar no es un parche.** MiniPlasma estuvo seis años en estado zombie. Un fix no está cerrado hasta que un test de regresión demuestra que el PoC original ya no funciona.
4. **El control nativo también es superficie.** El propio Defender fue el vehículo de la escalada; WinRE, el de los bypass de BitLocker. La defensa en profundidad no es un lujo: si un solo control cae —o se apaga sin avisar— no puede quedarte a ciegas.

Y un cierre sobre la naturaleza de la amenaza: esto no lo movió una APT con recursos, sino **una persona con conocimiento de insider y un rencor**. Su código, publicado por despecho, terminó en manos de bandas de ransomware. La ventana que abre un investigador enfadado es, para el defensor, indistinguible de la que abriría un adversario patrocinado —y llega con menos aviso—.

14 · Glosario

Términos técnicos que atraviesan el informe, en el sentido concreto en que se usan aquí.

TÉRMINO	QUÉ ES
TOCTOU (CWE-367)	*Time-Of-Check to Time-Of-Use*. Condición de carrera en la que un programa valida un recurso y luego actúa sobre él asumiendo que no cambió; el atacante lo cambia en el intervalo. Clase de vulnerabilidad central de la campaña.
Oplock	*Opportunistic lock* (variante *batch*). Mecanismo legítimo de caché de ficheros de Windows que permite pausar una operación; se abusa como «freno programable» para congelar a Defender en el instante crítico.
Junction / *reparse point*	Enlace de directorio de NTFS que redirige una ruta a otra. Un usuario sin privilegios puede crearlos en carpetas donde escribe (`%TEMP%`).
WER	*Windows Error Reporting*. Subsistema de reporte de errores; su tarea `QueueReporting` corre como SYSTEM y ejecuta `wermgr.exe` — vehículo recurrente de ejecución como SYSTEM en esta campaña.
LPE	*Local Privilege Escalation*. Elevación de privilegios local (típicamente hasta SYSTEM).
SYSTEM	La cuenta de mayor privilegio del sistema operativo Windows; el objetivo de las piezas de escalada.
WinRE	*Windows Recovery Environment*. Entorno de recuperación que corre en el arranque temprano, antes de las fronteras de seguridad normales; superficie de los bypass de BitLocker.
TxF	*Transactional NTFS*. Registro transaccional de NTFS cuyos logs procesa `autofstx.exe` en WinRE (vector de YellowKey).
SAM	*Security Account Manager*. Hive del registro con los hashes de credenciales locales; su lectura habilita robo de credenciales (T1003.002).
CTFMON	*Collaborative Translation Framework* / `ctfmon.exe`. Servicio de entrada de texto que corre como SYSTEM (superficie de GreenPlasma).
DACL	*Discretionary Access Control List*. Lista de control de acceso de un objeto; los symlinks de registro se usan para eludirla.
KEV	*Known Exploited Vulnerabilities*. Catálogo de CISA de vulnerabilidades con explotación confirmada; impone plazos de parcheo a agencias federales.
OOB	*Out-of-band*. Parche de emergencia fuera del ciclo mensual de Patch Tuesday.
WDAC / AppLocker	Tecnologías de *application allowlisting* de Windows. Frente a la suplantación por junction, solo son fiables las reglas por *publisher/hash*, no por ruta.
Tamper Protection	Función de Defender que dificulta —no impide— su manipulación.
TPM+PIN	Configuración de BitLocker que exige un PIN además del TPM; bloquea el escenario TPM-only que explota YellowKey.
Sigma / KQL / YARA	Formatos de detección: Sigma (regla portable de logs), KQL (consultas en Defender XDR/Sentinel), YARA (firmas sobre ficheros).

15 · Referencias

Fuentes públicas consultadas y verificadas. Los análisis originales por pieza, con sus reglas completas y el registro de actualizaciones, están en fennek.org.

- Huntress — *Nightmare-Eclipse Tooling Seen in Real-World Intrusion*: huntress.com
- Picus — *BlueHammer & RedSun (`CVE-2026-33825`) Explained*; *RoguePlanet: Anatomy of the Nightmare Eclipse Defender Zero-Day*: picussecurity.com
- Cyderes — *RoguePlanet: Windows Zero-Day Weaponizes Defender Quarantine Pipeline*: cyderes.com
- GuardSix — *Inside the Latest Chaotic-Eclipse Releases: MiniPlasma, GreenPlasma, YellowKey*: guardsix.com
- ThreatLocker — *MiniPlasma: Privilege escalation 0-day affects fully patched systems*; *GreatXML: Exploiting the WinRE trust boundary*: threatlocker.com
- Eclypsium — *YellowKey: The BitLocker Bypass Hidden in Windows Recovery*: eclypsium.com
- Help Net Security — *Defender vulnerabilities exploited (`CVE-2026-41091`, `CVE-2026-45498`)*; *YellowKey mitigation*; *RoguePlanet (`CVE-2026-50656`)*: helpnetsecurity.com
- SecurityWeek — *Microsoft Patches UnDefend and RedSun Zero-Days*; *BlueHammer Exploited in Ransomware Attacks*; *'GreatXML' Bypasses BitLocker*: securityweek.com
- BleepingComputer — *CISA: Windows BlueHammer flaw now exploited by ransomware gangs*; *Microsoft working on Defender patch for RoguePlanet*; *Nightmare Eclipse's July 14 mass reveal is off the table*: bleepingcomputer.com
- The Register — *Disgruntled 0-day hunter pledges 'bone shattering drop'*: theregister.com
- CISA — *Adds Known Exploited Vulnerabilities to Catalog*; KEV Catalog: cisa.gov/known-exploited-vulnerabilities-catalog
- NVD — fichas de cada CVE: `CVE-2026-33825` · `CVE-2026-41091` · `CVE-2026-45498` · `CVE-2026-45585` · `CVE-2026-45586` · `CVE-2026-50656` · `CVE-2020-17103` en nvd.nist.gov
- MITRE — CWE-367 (TOCTOU) · ATT&CK

© 2026 dust115 · fennek.org — TLP:CLEAR. Documento de síntesis defensiva; puede redistribuirse citando la fuente. Los nombres de producto y las marcas pertenecen a sus respectivos titulares.