Los atacantes lograron tomar el control de más de 400 paquetes alojados en el Arch User Repository (AUR) durante esta semana y modificaron sus scripts de compilación para instalar un ladrón de credenciales en cualquier sistema donde fueran construidos.

El malware consiste en un binario desarrollado en Rust diseñado para recopilar secretos y credenciales de desarrolladores. Cuando se ejecuta con privilegios de administrador, también puede desplegar un rootkit basado en eBPF para ocultar su presencia. Cabe destacar que el AUR es el repositorio comunitario de paquetes de Arch Linux y funciona de manera independiente de los repositorios oficiales de Arch, los cuales no resultaron afectados.

Si instalaste o actualizaste algún paquete desde AUR a partir del 11 de junio, es recomendable contrastarlo con las listas de paquetes comprometidos actualmente disponibles antes de considerar el sistema como seguro. La relación de paquetes afectados continúa creciendo y todavía no se considera definitiva.

Este incidente no explota una vulnerabilidad de software tradicional, sino que ataca directamente el modelo de confianza del ecosistema. Los paquetes comprometidos conservaron sus nombres, historial y reputación previa; únicamente fueron alteradas sus instrucciones de compilación.

La trampa estaba escondida en las recetas de construcción, mientras que el software distribuido aparentaba ser exactamente el mismo que los usuarios esperaban instalar. No hubo exploits, vulnerabilidades de día cero ni señales de que la infraestructura oficial de Arch Linux hubiera sido comprometida.

Los responsables de la campaña adoptaron paquetes abandonados, modificaron sus archivos de construcción y permitieron que fueran los propios usuarios quienes ejecutaran el código malicioso durante la instalación. La empresa Sonatype, que bautizó la operación como Atomic Arch, descubrió que los atacantes se centraron en proyectos huérfanos: paquetes cuyos mantenedores originales los habían abandonado, dejándolos disponibles para ser adoptados por cualquier persona.

Además, falsificaron los metadatos de los commits de Git para que las modificaciones parecieran realizadas por un mantenedor veterano. Posteriormente, un Usuario de Confianza de Arch Linux confirmó que la cuenta legítima de dicho mantenedor nunca fue comprometida.

Una vez obtenían el control de un paquete, modificaban su archivo PKGBUILD o su script .install para ejecutar el comando `npm install atomic-lockfile` durante la compilación. Esto descargaba el paquete malicioso desde npm junto con otros paquetes legítimos para pasar desapercibido. El paquete involucrado, `atomic-lockfile@1.4.2`, incorporaba un gancho de preinstalación que ejecutaba un binario ELF para Linux denominado `deps`. Al compilar el paquete, dicho binario se ejecutaba automáticamente.

Entre los ejemplos confirmados reportados a la lista de correo de Arch Linux se encuentran los paquetes `alvr` y `premake-git`.

## ¿Qué hace el malware?

El investigador independiente Whanos analizó el binario `deps` y determinó que se trata de un ladrón de credenciales desarrollado en Rust, orientado principalmente a estaciones de trabajo de desarrolladores y entornos de compilación.

La amenaza es capaz de recopilar:

- Cookies, tokens y almacenamiento local de navegadores basados en Chromium, como Chrome, Edge, Brave y otros.
- Datos de sesión de aplicaciones Electron, incluyendo Slack, Discord y Microsoft Teams.
- Tokens de GitHub, npm y HashiCorp Vault, además de credenciales relacionadas con OpenAI/ChatGPT y metadatos de cuentas.
- Claves SSH, archivos `known_hosts` e historiales de comandos de la terminal.
- Credenciales de Docker y Podman, así como perfiles VPN.

La información robada es enviada mediante HTTP al servicio temp.sh, mientras que la

comunicación con los servidores de comando y control se realiza a través de una dirección Onion de Tor utilizando un proxy local.

Para garantizar su persistencia, el malware instala un servicio systemd configurado con la opción `Restart=always`. Cuando dispone de privilegios de root, copia sus archivos en `/var/lib/` y crea una unidad de servicio dentro de `/etc/systemd/system/`. Si opera con permisos de usuario estándar, utiliza el directorio personal y crea una unidad en `~/.config/systemd/user/`. En ambos casos, el objetivo es permanecer activo tras reinicios del sistema.

Los primeros análisis exageraron el papel del rootkit eBPF. En realidad, este componente es opcional y solo se activa cuando el malware ya cuenta con privilegios elevados y las capacidades necesarias. No se utiliza para escalar privilegios. Cuando entra en funcionamiento, oculta procesos, nombres de procesos y sockets asociados al malware mediante mapas BPF denominados `hidden_pids`, `hidden_names` y `hidden_inodes`, además de bloquear intentos de depuración.

Este detalle modifica significativamente las recomendaciones de mitigación. Eliminar el paquete de AUR no es suficiente una vez que el malware se ha ejecutado. El gestor de paquetes puede eliminar archivos conocidos, pero no puede garantizar que el sistema quede completamente limpio después de que un componente con capacidades de rootkit haya tenido oportunidad de operar.

Asimismo, el binario despliega un segundo archivo relacionado con `monero-wallet-gui`, identificado durante el análisis como un posible criptominero que aún no ha sido estudiado en profundidad. La combinación de un ladrón de credenciales con un rootkit eBPF convierte esta amenaza en un incidente especialmente relevante.

## Alcance del incidente y segunda ola de ataques

El informe inicial de Sonatype identificó más de 20 paquetes secuestrados. Sin embargo, en menos de 24 horas, los rastreadores comunitarios y los debates en el canal aur-general de

Arch Linux habían contabilizado más de 400 paquetes afectados. Una lista maestra elaborada mediante búsquedas en el espejo Git del AUR elevó la cifra a aproximadamente 408 paquetes, mientras que otros listados consolidados continuaron aumentando.

Curiosamente, el paquete malicioso `atomic-lockfile` registraba apenas 134 descargas semanales en Socket antes de ser eliminado del registro npm. Esto indica que la verdadera superficie de exposición se encontraba en el proceso de construcción de paquetes desde AUR y no en las instalaciones directas desde npm.

Posteriormente se detectó una segunda campaña que utilizaba el comando `bun install js-digest`, distribuida desde otro conjunto de cuentas. Investigadores de la comunidad vincularon estas cuentas con el mismo publicador de npm responsable de `atomic-lockfile`. Esta nueva variante descargaba un binario diferente, identificado por un hash distinto y también catalogado como malicioso.

Todavía se está evaluando el alcance exacto de esta segunda oleada. Los primeros reportes hablaban de varias decenas de paquetes comprometidos, mientras que búsquedas posteriores en los repositorios espejo del AUR arrojaron cifras considerablemente superiores. Independientemente del número final, los expertos recomiendan verificar la presencia tanto de `atomic-lockfile` como de `js-digest`.

## Recomendaciones para los usuarios

Los mantenedores de Arch Linux están revirtiendo los commits maliciosos, bloqueando las cuentas involucradas y solicitando a la comunidad que continúe reportando paquetes sospechosos.

Dado que la lista de paquetes afectados aún no está completa, se recomienda:

- Revisar cualquier paquete AUR instalado o actualizado desde el 11 de junio utilizando las listas comunitarias y herramientas de detección disponibles.
- Buscar en historiales de compilación y cachés referencias a `npm install atomic-`

`lockfile`, `bun install js-digest` y a la ruta `src/hooks/deps`.

- Si se ejecutó alguno de los paquetes comprometidos, asumir que las credenciales del sistema han sido expuestas y proceder a rotar todas las credenciales potencialmente afectadas, incluyendo sesiones de navegador, claves SSH, tokens de GitHub y npm, sesiones de Slack, Teams y Discord, credenciales de Vault, Docker, Podman y claves de servicios en la nube.
- Investigar posibles mecanismos de persistencia revisando servicios systemd desconocidos, tanto a nivel de sistema como de usuario, además de archivos sospechosos en `/var/lib/`.
- Inspeccionar `/sys/fs/bpf/` para detectar la presencia de los mapas `hidden_pids`, `hidden_names` y `hidden_inodes`.
- Analizar conexiones salientes hacia la red Tor y servicios de carga de archivos.

Si el paquete comprometido fue ejecutado con privilegios de administrador, la recomendación es asumir que el rootkit está presente y reinstalar completamente el sistema desde medios confiables. No existe una forma fiable de garantizar la integridad del equipo en esas circunstancias.

Como medida preventiva a futuro, se aconseja revisar cuidadosamente los archivos PKGBUILD y cualquier script `.install` antes de compilar un paquete, especialmente cuando se trate de proyectos recientemente adoptados o que hayan mostrado actividad repentina después de largos periodos de inactividad. Si las instrucciones de compilación no son comprendidas en su totalidad, lo más prudente es evitar la instalación.

Para labores de detección, el hash SHA-256 del principal componente malicioso es:

`6144d433f8a0316869877b5f834c801251bbb936e5f1577c5680878c7443c98b`

El conjunto completo de indicadores de compromiso, incluida la dirección Onion utilizada para el comando y control, se encuentra documentado en el análisis publicado por ioctl.fail.

No es la primera vez que una táctica de adopción maliciosa afecta al ecosistema AUR. Un

método similar fue empleado contra un paquete abandonado de visualización de PDF en 2018. Sin embargo, la versión observada en 2026 se distingue por su escala, formando parte de una tendencia creciente de ataques a la cadena de suministro que buscan heredar la confianza de proyectos abandonados en lugar de recurrir a técnicas tradicionales como el typosquatting.

Actualmente no se ha asignado un identificador CVE para este incidente. Sonatype realiza su seguimiento bajo el código Sonatype-2026-003775, con una puntuación CVSS de 8.7.

En última instancia, el ataque tuvo éxito porque el modelo de confianza del AUR sigue apoyándose principalmente en el nombre y el historial de un paquete, más que en la identidad de quien lo mantiene en la actualidad. A partir de ahora, cualquier paquete recientemente adoptado o que incorpore nuevos scripts de instalación de forma inesperada debería analizarse con el mismo nivel de desconfianza que un paquete publicado por un desarrollador desconocido.