Investigadores en ciberseguridad han revelado detalles sobre un marco de botnet para dispositivos del Internet de las Cosas (IoT), hasta ahora no documentado, denominado TuxBot v3 Evolution, el cual presenta indicios de haber sido desarrollado con la asistencia de un modelo de lenguaje de gran escala (LLM), aunque con resultados limitados en cuanto a su funcionamiento.

"Si bien la inteligencia artificial respondió a la solicitud de generar el código de la botnet, también incorporó un aviso de seguridad que el desarrollador olvidó eliminar antes de distribuir el software. Aunque el LLM contribuyó claramente a la construcción de la botnet, varias funciones presentes en las muestras analizadas no operaban correctamente", señaló Palo Alto Networks Unit 42. *"Un proceso de revisión manual del código habría permitido corregir estos errores, por lo que no se descarta la existencia de versiones más depuradas del malware circulando actualmente."*

De acuerdo con la empresa de ciberseguridad, el framework está compuesto por diversos elementos: un agente bot desarrollado en C con capacidad de compilación cruzada para múltiples arquitecturas (ARM, MIPS, MIPSEL, MIPS64, x86_64, PowerPC y RISC-V); un servidor de comando y control (C2) implementado en Go que incorpora un panel para ofrecer ataques DDoS como servicio; una máquina virtual personalizada para exploits; una infraestructura de pruebas basada en Docker; y un sistema automatizado de compilación.

El agente bot está diseñado para realizar ataques de fuerza bruta contra servicios Telnet utilizando un conjunto de 1.496 combinaciones de credenciales. Asimismo, integra código de explotación dirigido a más de 30 familias de dispositivos IoT mediante vulnerabilidades previamente conocidas. La comunicación con el servidor C2 se lleva a cabo a través de un canal TCP cifrado, complementado con un algoritmo de generación de dominios basado en SHA512 (DGA), un protocolo de comunicación entre pares (P2P) con comandos firmados mediante Ed25519, Internet Relay Chat (IRC), consultas DNS TXT y sondeos HTTP como mecanismo de respaldo.

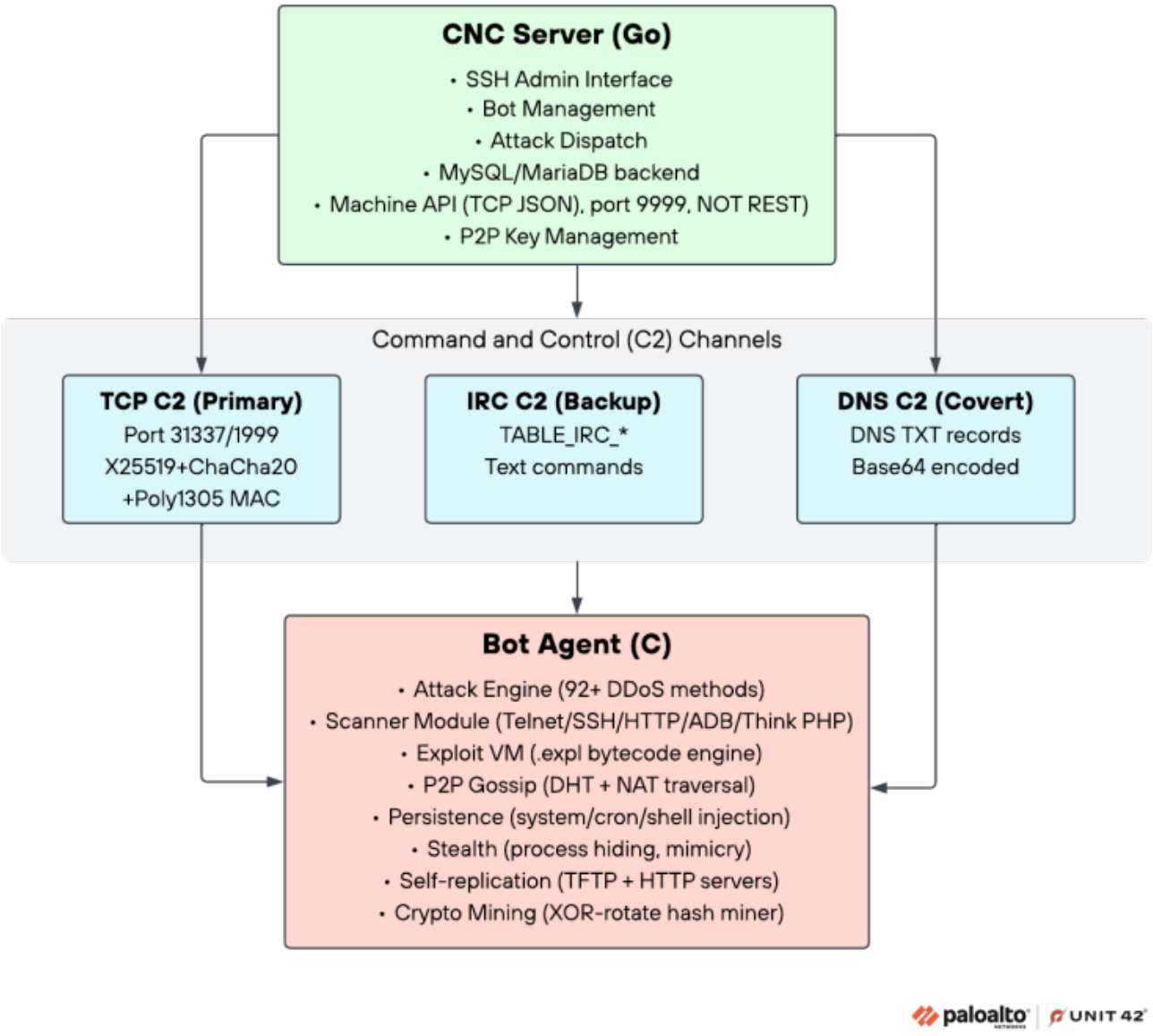
El origen de este framework modular se ha relacionado con tres botnets distintas: Mirai, AISURU y Wuhan. Además, parte de sus funcionalidades fueron adaptadas desde la

herramienta de código abierto MHDDoS, desarrollada en Python para ataques distribuidos de denegación de servicio (DDoS). Una muestra del malware fue cargada en la plataforma VirusTotal el 20 de enero de 2026, lo que indica que ha estado activa durante más de seis meses. Las evidencias recopiladas también sugieren que el desarrollo comenzó aproximadamente un año antes, cuando el autor clonó el repositorio de MHDDoS desde GitHub.

"Según la descripción del framework, el desarrollador de TuxBot creó lo que denomina una plataforma profesional de comando y control con un panel administrativo multiusuario, despliegue automatizado y capacidades modulares para la ejecución de ataques", afirmaron los investigadores Chris Navarrete, Asher Davila y Doel Santos.

El componente C2, desarrollado en Go, utiliza tres puertos TCP diferenciados para gestionar las conexiones entrantes:

- Puerto TCP 1999 (o 31337): destinado al envío cifrado de comandos hacia los bots conectados.
- Puerto TCP 2222: proporciona una consola interactiva para los operadores mediante SSH.
- Puerto TCP 9999: ofrece una interfaz basada en JSON para el acceso programático.



Una vez iniciado, el botnet ejecuta una secuencia predefinida de inicialización que incluye las siguientes acciones:

- Obtención de la dirección del servidor C2 mediante una arquitectura multinivel compuesta por un canal principal y cinco mecanismos alternativos.

- Implementación de técnicas anti-depuración y anti-máquina virtual para detectar herramientas de análisis en ejecución.
- Ocultamiento del nombre del proceso.
- Configuración de mecanismos de persistencia.
- Ejecución de distintos submódulos destinados a lanzar ataques DDoS, finalizar procesos competidores, establecer canales C2 mediante IRC, HTTP, DNS y P2P, ejecutar escáneres para Telnet, SSH, HTTP y Android Debug Bridge (ADB), desplegar un proxy SOCKS5 e iniciar un módulo reservado para minería de criptomonedas.

En particular, el escáner HTTP es capaz de mantener hasta 128 conexiones simultáneas con el propósito de localizar interfaces web vulnerables. La persistencia del malware se consigue mediante la creación de un servicio systemd, la incorporación de tareas programadas en cron y un proceso watchdog encargado de garantizar que TuxBot continúe ejecutándose en el sistema comprometido.

"Diversos archivos contienen, de forma literal, el razonamiento interno generado por el LLM durante el proceso de desarrollo, conservado dentro de los comentarios del código. Estos comentarios reflejan la cadena de razonamiento del modelo mientras realizaba tareas de adaptación, incluyendo interrupciones, decisiones y referencias al 'usuario', es decir, al desarrollador que interactuó con el modelo", indicó Unit 42.

Aunque TuxBot v3 Evolution continúa en fase de desarrollo, las funciones que ya se encuentran operativas, junto con la utilización de inteligencia artificial para acelerar su implementación, evidencian cómo un único desarrollador puede construir una plataforma compleja que integra múltiples canales de comando y control, una máquina virtual personalizada para exploits y un panel DDoS implementado en Go.

"La infraestructura compartida con Kaitori v3.9 y las herramientas de AISURU sitúan al operador de TuxBot dentro del ecosistema Keksec. Este grupo es conocido por mantener simultáneamente varias variantes de botnets dirigidas a dispositivos IoT. Todo indica que TuxBot constituye una nueva incorporación a ese conjunto de herramientas. Su objetivo parece ser superar las capacidades habituales de las variantes derivadas de Mirai mediante

un C2 cifrado, un algoritmo DGA y un sistema modular de exploits, aunque este último todavía no funciona correctamente en la versión recuperada”, concluyó Unit 42.

Esta revelación se produce tras la aparición de otras dos botnets, RustDuck y AryStinger, las cuales han dirigido sus ataques contra routers, cámaras IP, dispositivos Android TV y servidores con configuraciones de seguridad deficientes, incorporándolos a redes maliciosas utilizadas tanto para ejecutar ataques de denegación de servicio contra plataformas en línea como para realizar actividades de reconocimiento.