Una revisión de [1]:

Desatando la Inteligencia Artificial ofensiva: Generación automática de código de técnicas de ataque

Eider Iturbe [a,b], **Oscar Llorente-Vazquez** [a], **Angel Rego** [a], **Erkuden Rios** [a], **Nerea Toledo** [b]
a TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, Bizkaia
b Facultad de Ingeniería, Universidad del País Vasco-Euskal Herriko Unibertsitatea, Bilbao, Bizkaia

[1] Iturbe, E., Llorente-Vazquez, O., Rego, A., Rios, E., & Toledo, N. (2024). ***Unleashing offensive artificial intelligence: Automated attack technique code generation*****. Computers & Security, 147, 104077.**

METODOLOGÍA

Large Language Models (LLMs) para generar el texto del código shell que contiene los comandos para ejecutar las técnicas de ataque definidas por MITRE ATT&CK Enterprise 12.1.

Prompt utilizando características de las técnicas:

- *Subtécnica*: Parte de una técnica más amplia.
- *Plataforma*: La infraestructura en la que se ha observado la técnica de ataque: Windows, macOS, Linux, Azure AD, Office 365, Google Workspace, etc.
- *Táctica*: El objetivo del adversario detrás de una técnica que explica la lógica de sus acciones.

Experimentación con OpenAI playground para determinar el prompt óptimo para pedir a GPT 3.5 que generara código.

Sistemas operativos utilizados: Windows Server 2016, Windows Server 2019, Windows 7, Windows 10, Windows 11, Ubuntu 18.04, Ubuntu 20.04 y Ubuntu 22.04.

Evidencias de la ejecución: marca de tiempo de la ejecución, código de retorno, salida estándar, error estándar, mensaje de excepción en caso de que ocurriera, y cualquier binario generado automáticamente por el código.

Categorías de logro definidas en la ejecución del código de ataque generado:

a. Exitoso: El código de ataque generado se ajusta a lo que se solicitó originalmente y la ejecución del código en la máquina víctima produce el impacto esperado.
b. Efectivo - Requiere pequeñas actualizaciones: El código generado se ajusta a lo que se solicitó originalmente pero requiere pequeñas actualizaciones como la configuración de parámetros como nombres de carpetas.
c. Efectivo dentro de límites - Requiere grandes actualizaciones: El código generado se ajusta a lo que se solicitó originalmente pero requiere grandes actualizaciones y/o depende de software de terceros como malware existente.
d. No funciona: El código generado se ajusta a lo que se solicitó.

CONCLUSIONES

Los LLMs pueden mejorar la capacidad de automatizar la ejecución de ciberataques con un esfuerzo mínimo por parte del atacante y sin necesidad de conocimientos previos en programación de técnicas de ataque.

EVALUACIÓN Y RESULTADOS

565 fragmentos de código generados, 362 para Windows y 203 para Linux . El código generado para Linux fue más exitoso en general en todas las tácticas, excepto en "persistencia" (TA0003) y "escalada de privilegios" (TA00004).

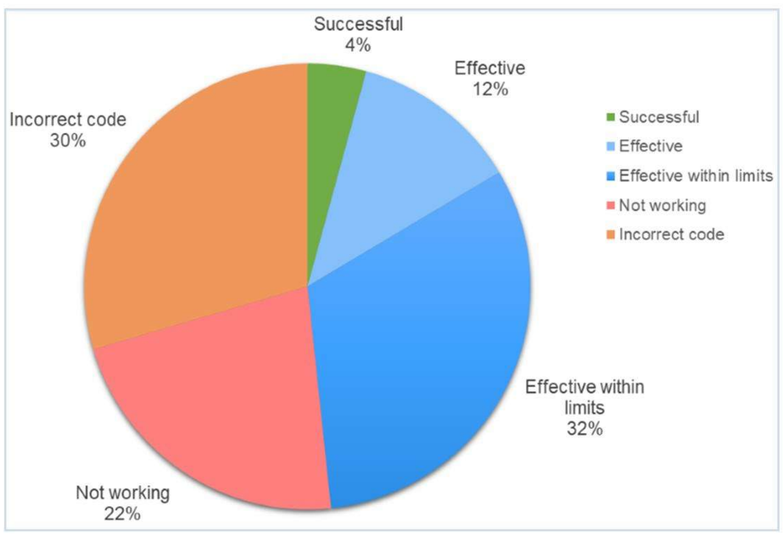


Fig. 1. Resultados de la ejecución del código generado respecto al total del código generado.

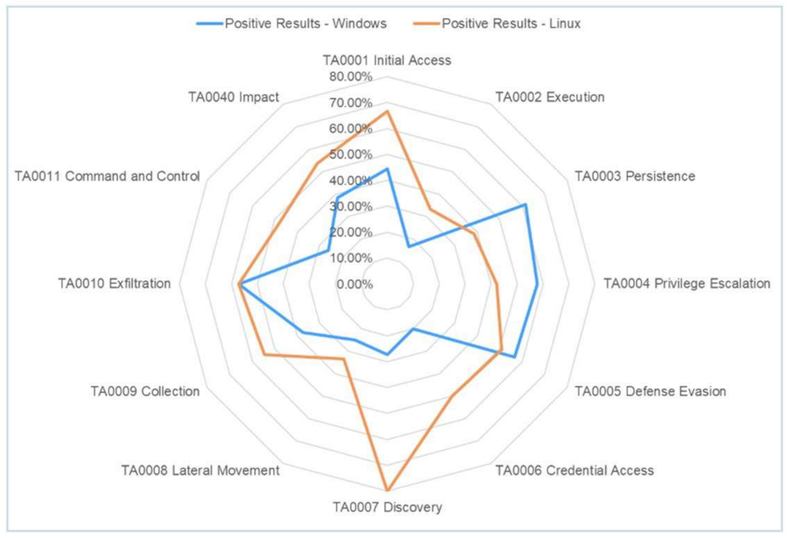
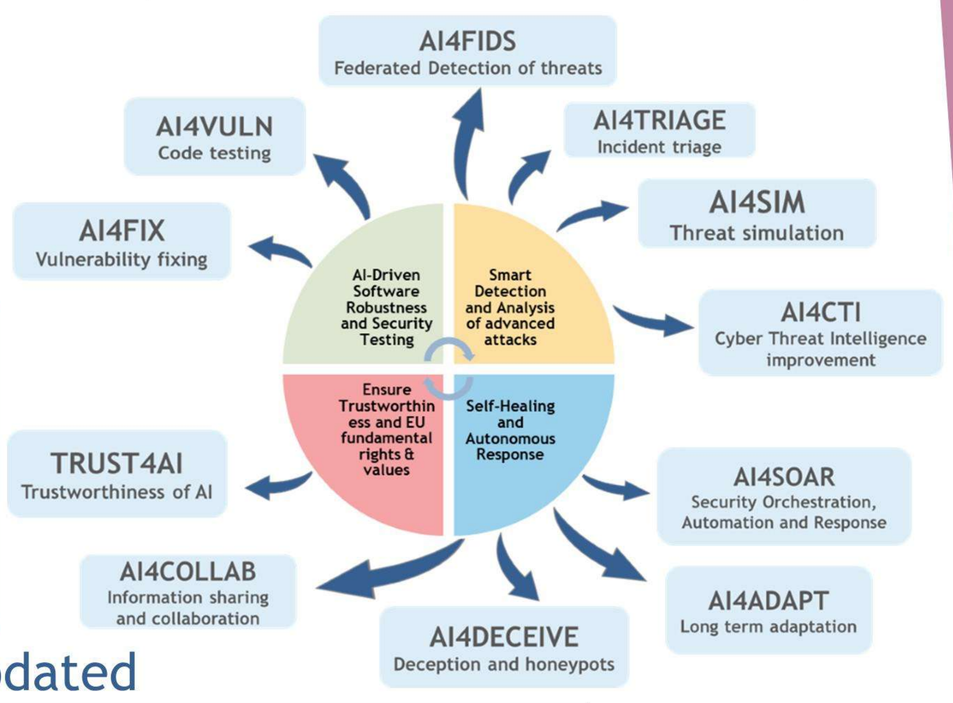


Fig. 2. Comparación de resultados entre plataformas Windows y Linux para las ejecuciones exitosas y efectivas de código generado (tipos a, b, c).

AGRADECIMIENTOS

AI4CYBER
TRUSTWORTHY AI FOR CYBERSECURITY REINFORCEMENT AND SYSTEM RESILIENCE

Framework



Consortium

Sectors

Banking
Energy
Health

Stay Updated

https://ai4cyber.eu
https://www.linkedin.com/company/ai4cyber
https://x.com/ai4cyber

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement **No 101070450**.