Los investigadores en ciberseguridad han descubierto una grave vulnerabilidad en la biblioteca Vanna.AI que podría ser explotada para lograr una ejecución remota de código mediante técnicas de inyección de instrucciones.

La vulnerabilidad, identificada como CVE-2024-5565 (con un puntaje CVSS de 8.1), está vinculada a un caso de inyección de instrucciones en la función «ask» que podría ser utilizada para engañar a la biblioteca y ejecutar comandos arbitrarios, según la firma de seguridad en la cadena de suministro JFrog.

Vanna es una biblioteca de aprendizaje automático basada en Python que permite a los usuarios interactuar con su base de datos SQL para obtener información *«simplemente haciendo preguntas»* (también conocidas como instrucciones) que se traducen en una consulta SQL equivalente utilizando un modelo de lenguaje grande (LLM).

La rápida adopción de modelos de inteligencia artificial (IA) generativa en los últimos años ha puesto de relieve los riesgos de explotación por parte de actores malintencionados, quienes pueden aprovechar las herramientas proporcionando entradas adversariales que evaden los mecanismos de seguridad incorporados.

Una clase destacada de ataques es la inyección de instrucciones, que se refiere a un tipo de evasión de las restricciones de la IA que puede utilizarse para ignorar las barreras establecidas por los proveedores de LLM para prevenir la producción de contenido ofensivo, dañino o ilegal, o para llevar a cabo instrucciones que violan el propósito previsto de la aplicación.

Estos ataques pueden ser indirectos, donde un sistema procesa datos controlados por terceros (como correos electrónicos entrantes o documentos editables) para lanzar una carga maliciosa que conduce a una evasión de la IA.

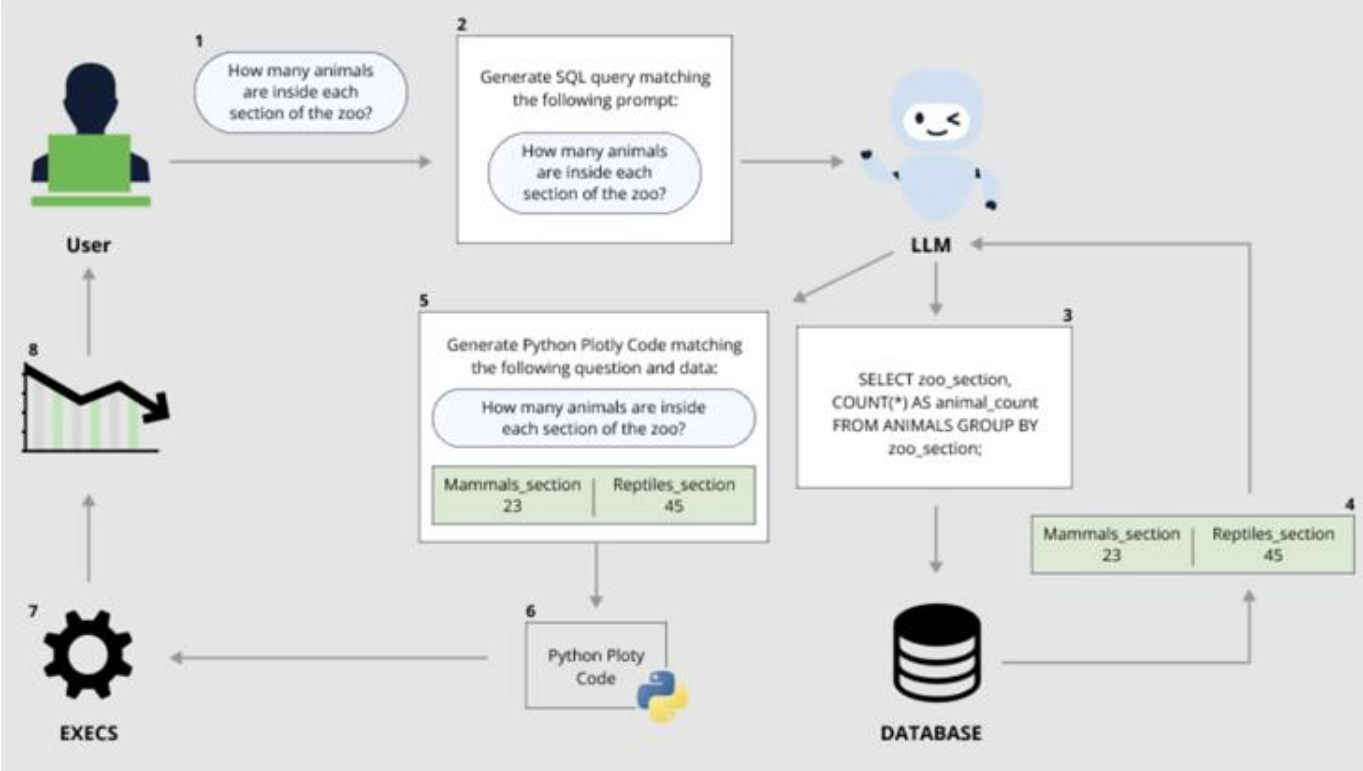
También pueden tomar la forma de una evasión de múltiples intentos o de múltiples turnos (también conocida como Crescendo), donde el operador *«comienza con un diálogo inofensivo y gradualmente dirige la conversación hacia el objetivo prohibido».*

Este enfoque puede extenderse aún más para realizar otro nuevo ataque de evasión conocido como Llave Maestra.

> *«Esta técnica de evasión de la IA funciona utilizando una estrategia de múltiples turnos (o múltiples pasos) para hacer que un modelo ignore sus barreras. Una vez que las barreras son ignoradas, un modelo no podrá diferenciar entre solicitudes maliciosas o no autorizadas y cualquier otra»,* dijo Mark Russinovich, director de tecnología de Microsoft Azure

La Llave Maestra también se diferencia de Crescendo en que, una vez que la evasión es exitosa y las reglas del sistema cambian, el modelo puede generar respuestas a preguntas que de otro modo estarían prohibidas, independientemente de los riesgos éticos y de seguridad involucrados.

> *«Cuando la evasión de la Llave Maestra tiene éxito, un modelo reconoce que ha actualizado sus directrices y posteriormente cumplirá con las instrucciones para producir cualquier contenido, sin importar cuánto viole sus directrices originales de IA responsable,»* explicó Russinovich.



> *«A diferencia de otros jailbreaks como Crescendo, donde se debe consultar a los modelos sobre tareas de manera indirecta o con codificaciones, Skeleton Key pone a los modelos en un modo en el que el usuario puede solicitar tareas directamente. Además, la salida del modelo parece estar completamente sin filtrar y revela el alcance del conocimiento del modelo o su capacidad para producir el contenido solicitado».*

Los últimos descubrimientos de JFrog, también divulgados de manera independiente por Tong Liu, muestran cómo las inyecciones de prompts podrían tener graves impactos, particularmente cuando están vinculadas a la ejecución de comandos.

CVE-2024-5565 se aprovecha del hecho de que Vanna facilita la generación de texto a SQL para crear consultas SQL, que luego se ejecutan y se presentan gráficamente a los usuarios utilizando la biblioteca de gráficos Plotly.

Esto se logra mediante una función de «preguntar», por ejemplo, vn.ask(*«¿Cuáles son los 10 principales clientes por ventas?»*), que es uno de los principales puntos finales de la API que permite la generación de consultas SQL para ser ejecutadas en la base de datos.

El comportamiento mencionado, junto con la generación dinámica del código Plotly, crea una vulnerabilidad de seguridad que permite a un actor malicioso enviar un prompt especialmente diseñado que incrusta un comando para ser ejecutado en el sistema subyacente.

> *«La biblioteca Vanna utiliza una función de prompt para presentar al usuario resultados visualizados; es posible alterar el prompt usando inyección de prompts y ejecutar código Python arbitrario en lugar del código de visualización previsto»,* dijo JFrog.
>
> *«Específicamente, permitir la entrada externa al método 'ask' de la biblioteca con 'visualize' configurado en True (comportamiento predeterminado) lleva a la ejecución remota de código.»*

Tras la divulgación responsable, Vanna ha emitido una guía de fortalecimiento que advierte a los usuarios que la integración de Plotly podría ser utilizada para generar código Python arbitrario y que los usuarios que expongan esta función deberían hacerlo en un entorno aislado.

> *«Este descubrimiento demuestra que los riesgos del uso generalizado de GenAI/LLMs sin una gobernanza y seguridad adecuadas pueden tener implicaciones drásticas para las organizaciones»,* dijo Shachar Menashe, director senior de investigación de seguridad en JFrog, en un comunicado.

> *«Los peligros de la inyección de prompts aún no son ampliamente conocidos, pero son fáciles de ejecutar. Las empresas no deberían confiar en el pre-prompting como un mecanismo de defensa infalible y deberían emplear mecanismos más robustos cuando interfieren LLMs con recursos críticos como bases de datos o generación de código dinámico.»*