Gemini CLI: A New Addition to Kali Linux 2025.3

In this presentation we will cover Gemini CLI, one of the new tools in Kali Linux 2025.3. This powerful command-line interface (CLI) allows users to directly interact with the Gemini Protocol, fetching resources from Gemini servers. It empowers developers, security researchers, and practitioners with streamlined access to Gemini content. Additionally the tool will allow security practitioners

For clarity, Google developed Gemini-CLI, not Kali Linux. This tool significantly advances how security professionals interact with the Gemini Protocol, offering diverse command-line options to customize behavior and integrate seamlessly into existing workflows.

Understanding the Gemini Protocol

The Gemini Protocol is a simple, text-based protocol focusing on privacy and security. It uses a single connection to transfer data, reducing latency and enhancing performance compared to complex protocols like HTTP. Gemini-CLI offers a straightforward way to access and manipulate Gemini resources, with various command-line options for customization.

Pseudonymous developer Solderpunk, introduced Gemini in 2019 as an application-layer internet communication protocol. Designed as a middle ground between Gopher's minimalism and HTTP's complexity, Gemini streamlines accessing and publishing online content.

Operating on TCP port 1965, the protocol mandates Transport Layer Security (TLS) encryption for all connections, ensuring secure communication by default. Unlike the modern web with its heavy reliance on JavaScript, tracking mechanisms, and resource-intensive applications, Gemini exclusively delivers text-based content in the simplest way possible.

TCP Port 1965

Standard communication port for Gemini Protocol

TLS Encryption

Mandatory security for all connections

Text-Based

Simple content delivery without complexity

Core Features of Gemini-CLI

Gemini-CLI provides core features and tools that enables access to local information, execute commands, and interact with the Internet. It reads files, lists directories, and runs shell commands with safety measures. Additionally, it fetches content from URLs and modifies system files. These tools enable Gemini to provide more accurate and relevant responses based on real-time or local data.

Access Local Information

Tools enable Gemini to access your local file system, read file contents, list directories, and efficiently navigate your system structure.

Execute Commands

Tools like `run_shell_command` allow Gemini to execute shell commands with appropriate safety measures and require user confirmation for sensitive operations.

Interact with the Web

Tools fetch content from URLs, enabling seamless integration between local operations and web-based resources.

Take Actions

Tools modify files, write new files, and perform other system actions, typically with safeguards to prevent unintended changes.

Ground Responses

By fetching real-time or specific local data, Gemini's responses become more accurate, relevant, and grounded in your operational context.

How Gemini CLI Tools Work

Gemini CLI tools operate through a sophisticated workflow, ensuring accurate and secure interactions. You begin by providing a prompt to the Gemini CLI. The CLI sends this prompt to the core, which then communicates with the Gemini API to retrieve available tools and their descriptions.

1

Prompt Submission

User provides a natural language prompt to the CLI.

2

API Communication

The core communicates with the Gemini API for tool descriptions.

3

Model Analysis

The Gemini model analyzes the request and identifies necessary tools.

4

Tool Execution

The core validates and executes the tool, requiring user confirmation for sensitive operations.

5

Response Delivery

The model formulates an answer using the tool's output and displays the result.

If the Gemini model determines a tool is necessary, it requests execution with specific parameters. The core validates this request, executes the tool (after user confirmation for sensitive operations), and returns the output to the model. The model then formulates its answer using this output and presents it to you.

Security Features

The Gemini protocol includes robust security features, safeguarding your interactions. This comprehensive framework protects sensitive data and prevents unauthorized access.

1 API Key Management

Gemini-cli enables secure API key management. Users store their API keys in environment variables or configuration files, preventing hard-coding sensitive information directly into scripts.

2 Encryption

Gemini-cli does not directly handle encryption but integrates with external tools and practices. Users can encrypt sensitive data, such as configuration files, using tools like GPG or OpenSSL.

3 Rate Limiting

Gemini-cli adheres to Gemini API rate limits, preventing abuse and ensuring fair usage. This maintains API stability and security.

4 Secure Communication

Gemini-cli communicates with the Gemini API via HTTPS, encrypting all data transmitted between the client and server.

5 Access Control

Users can set up fine-grained access controls on their Gemini API keys, limiting the actions a specific key can perform. This adds an extra layer of security, restricting potential damage in case of a key compromise.

Customization Features

Gemini prompts offer several customization features, allowing users to tailor the tool to their specific needs and workflow requirements.

Configuration Files

Gemini-cli supports configuration files for specifying settings and parameters. This enables easy customization without modifying source code.

Environment Variables

Users can override default settings using environment variables, offering flexible configuration for different environments (e.g., development, testing, production).

Command-Line Options

Gemini-cli provides a rich set of command-line options. These allow users to customize tool behavior per command, including specifying API endpoints and output formats.

Scripting and Automation

Designed for easy integration, Gemini-cli supports scripting and automation workflows. Users can write custom scripts to automate tasks like placing orders, checking balances, or retrieving market data.

Plugins and Extensions

While Gemini-cli does not directly support plugins, users can extend its functionality by writing custom scripts or integrating it with other tools and services.

Output Formats

Gemini-cli supports various output formats, including JSON, CSV, and plain text. This allows users to choose the format best suited for further processing or analysis.

Configuration Examples

Configuring Gemini-cli is crucial to maximize its potential. These practical examples demonstrate configuration file setup and command-line usage.

Example Configuration File

yaml Copy code

```yaml
api_key: "your_api_key_here"
api_secret: "your_api_secret_here"
base_url: "https://api.gemini.com"
rate_limit: 120
output_format: "json"
```

This configuration file structures settings for optimal Gemini-cli performance, including API endpoints, authentication credentials, and output preferences.

Example Command-Line Usage

sh Copy code

```sh
gemini-cli --config path/to/config.yaml --endpoint v1/orders --output-format csv
```

This example shows how to specify a configuration file, API endpoint, and output format using command-line options. It demonstrates the flexibility of combining configuration files with runtime parameters.

Leveraging these security and customization features ensures secure interactions with the Gemini API, tailored to specific needs.

Complete Gemini Toolkit Overview

The Gemini Toolkit is a comprehensive suite of tools for interacting with the Gemini API, facilitating cryptocurrency trading and management. It includes command-line interfaces (CLIs), libraries, and scripts to automate and streamline interactions with the Gemini exchange.

1 Gemini-cli

This command-line interface interacts directly with the Gemini API. Users can perform tasks like placing orders, checking account balances, and retrieving market data.

2 Gemini Python Library

This Python library offers a programmatic interface to the Gemini API. Use it to build custom applications, scripts, and automation workflows.

3 Gemini JavaScript Library

This JavaScript library interacts with the Gemini API, ideal for web applications and Node.js scripts.

4 REST API Documentation

Comprehensive documentation provides details on all available endpoints, request/response formats, and usage examples.

5 Sample Scripts

A collection of sample scripts and examples demonstrates how to effectively use the Gemini Toolkit for various tasks.

Key Toolkit Features

Order Management

Efficiently place, cancel, and retrieve order information. Supports market, limit, and other order types for comprehensive trading control.

Account Management

Check account balances, retrieve transaction history, and manage API keys securely.

Market Data

Access real-time and historical market data, including price ticks, order books, and trade history.

WebSocket API

Stream real-time data for instant market updates, order book changes, and trade executions.

Practical Usage Examples

The below examples demonstrate both CLI and Python library usage.

Some functions may require the use of an Gemini API key for proper functioning. The below URL can be utilized to obtain an API key for your use

https://aistudio.google.com/app/api-keys

Using Gemini-cli

This example demonstrates how to place a limit order using Gemini-cli:

sh Copy code

```sh
gemini-cli order new --symbol BTCUSD --amount 0.01 --price 50000 --side buy --type limit
```

This command places a limit order to buy 0.01 BTC at $50,000. Its straightforward structure makes Gemini-cli accessible to both beginners and advanced users.

Using Gemini Python Library

This example demonstrates how to retrieve account balances using the Gemini Python Library:

python Copy code

```python
from gemini import GeminiAPI

api_key = "your_api_key_here"
api_secret = "your_api_secret_here"

gemini = GeminiAPI(api_key, api_secret)
balances = gemini.get_balances()

for balance in balances:
    print(f"Currency: {balance['currency']}, Available: {balance['available']}, Reserved: {ba
```

This code retrieves and prints account balances for each currency. The Python library offers a programmatic approach, ideal for developers building custom applications.

Cybersecurity Use Cases

Gemini CLI integrates seamlessly into the Kali Linux environment, providing diverse cybersecurity solutions. It enhances security operations, from reconnaissance to incident response, by automating complex tasks and delivering intelligent insights. This empowers security professionals to work more efficiently and effectively across various scenarios.

01

Automated Reconnaissance

Automate target network reconnaissance to identify potential entry points for penetration testing.

02

Vulnerability Scanning

Identify and scan web application and system vulnerabilities using automated tools.

03

Exploitation & Post-Exploitation

Efficiently exploit identified vulnerabilities and gather critical post-exploitation data.

04

Incident Response

Quickly identify the source and scope of security breaches to facilitate rapid response.

05

Compliance & Auditing

Ensure regulatory compliance through automated security audits and assessments.

06

Threat Hunting

Proactively hunt for and neutralize potential threats within the network environment.

07

Red Team Operations

Simulate advanced persistent threat attacks to rigorously test organizational defenses.

Use Case 1: Automated Reconnaissance

Scenario

A security analyst gathers information about a target network to identify penetration testing entry points.

Benefit

Automating reconnaissance allows the analyst to focus on data analysis rather

1 Initial Setup

The analyst opens the Kali Linux terminal and initializes the Gemini CLI.

2 Natural Language Prompt

The analyst types, "Perform reconnaissance on the target IP range 192.168.1.0/24."

3 Automated Tasks

Gemini CLI runs tools like nmap, whois, and theHarvester to gather target network information.

4 Results

The CLI compiles a comprehensive report, highlighting open ports, running services, and potential vulnerabilities.

Use Case 2: Vulnerability Scanning

Security teams use Gemini CLI to efficiently identify web application vulnerabilities. It automates scanning tools and consolidates findings into actionable reports, streamlining vulnerability assessment.

1 Initial Setup

A team member opens the terminal and launches the Gemini CLI.

2 Natural Language Prompt

The team member inputs, "Scan the web application at http://example.com for vulnerabilities."

3 Automated Scan

Gemini CLI executes tools like Nikto, OWASP ZAP, and Nessus, performing a comprehensive vulnerability scan.

4 Detailed Report

The CLI generates a report detailing identified vulnerabilities, their severity, and remediation steps.

Benefit: This enables teams to rapidly identify, prioritize, and remediate critical security issues, significantly reducing the time to secure web applications.

Use Case 3: Exploitation and Post-Exploitation

Scenario

A penetration tester identifies a vulnerable service on a target system and needs to exploit it for access. This use case demonstrates how Gemini CLI streamlines exploitation and automates post-exploitation data gathering.

1 Initial Setup

The tester opens the terminal and launches the Gemini CLI.

2 Natural Language Prompt

The tester types, "Exploit the identified vulnerability on the target IP 192.168.1.100."

3 Automated Tasks

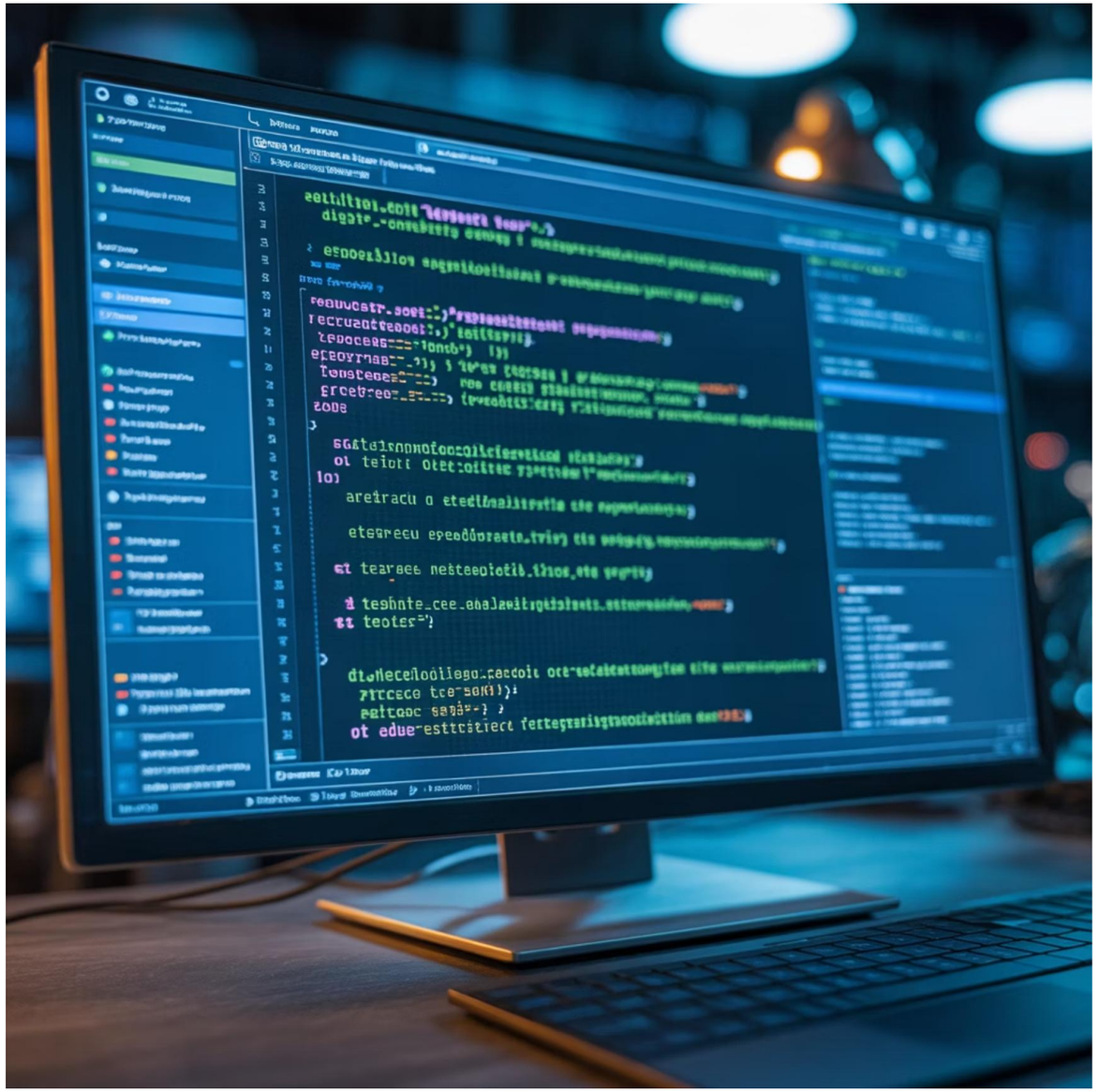
Gemini CLI uses tools like Metasploit to exploit the vulnerability and gain system access.

4 Post-Exploitation

The CLI automatically runs post-exploitation modules, gathering additional information like user credentials and system configurations.

5 Results

The tester receives a detailed report outlining the exploitation process and gathered information.



> "The tester efficiently exploits vulnerabilities and gathers post-exploitation data, streamlining

Use Case 4: Incident Response

A security incident occurred, and the response team must quickly identify its source and extent. Incident response demands speed, and Gemini CLI automates rapid assessment and threat containment.

1 Initial Setup

The incident response team member opens the terminal and launches Gemini CLI.

2 Natural Language Prompt

The team member types: "Investigate the security incident on the server with IP 192.168.1.50."

3 Automated Tasks

Gemini CLI leverages tools like Splunk, ELK Stack, and Wireshark to analyze logs, network traffic, and system events.

4 Results

The CLI generates a report detailing the breach's source, affected systems, and potential data exfiltration.

Benefit: Rapidly identify and contain breaches to minimize damage and downtime. This capability is crucial for limiting security incident impact and preventing further compromise.

Use Cases 5-7: Advanced Security Operations

1

Use Case 5: Compliance and Auditing

Scenario: Organizations must ensure regulatory compliance and conduct regular security audits.

Process: A compliance officer prompts Gemini CLI: "Perform a PCI-DSS compliance audit." Gemini CLI then uses tools like OpenVAS and Nessus to comprehensively audit the organization's systems and networks.

Benefit: Automated, comprehensive auditing ensures continuous regulatory compliance, reducing penalties and data breaches.

2

Use Case 6: Threat Hunting

Scenario: Security analysts proactively hunt for potential network threats.

Process: An analyst prompts Gemini CLI: "Hunt for indicators of compromise (IOCs) on the network." Gemini CLI analyzes network traffic and system logs for IOCs using tools like Splunk, ELK Stack, and Suricata.

Benefit: Proactively identify and mitigate threats, enhancing the organization's security posture before incidents occur.

3

Use Case 7: Red Team Operations

Scenario: A red team simulates advanced persistent threat (APT) attacks to test organizational defenses.

Process: A red team member prompts Gemini CLI: "Simulate an APT attack on the target network." Gemini CLI combines various tools and techniques to simulate APT attacks, including phishing, malware deployment, and lateral movement.

Benefit: Identify and address weaknesses in security defenses, improving overall resilience to real-world threats.

Gemini Command Line Switches

Gemini CLI offers an extensive array of command-line switches and options, providing fine-grained control over its behavior. Understanding these options is essential for maximizing the tool's effectiveness in your specific use case.

Core Options

- **-m, --model:** Specify the model.
- **-p, --prompt:** Provide a prompt (appended to stdin).
- **-i, --prompt-interactive:** Execute a prompt and continue in interactive mode.
- **-s, --sandbox:** Run in sandbox mode.
- **--sandbox-image:** Specify the sandbox image URI.
- **-d, --debug:** Enable debug mode (default: false).
- **-a, --all-files:** Include all files in context (default: false).

Advanced Options

- **--show-memory-usage:** Display memory usage in the status bar.
- **-y, --yolo:** Automatically accept all actions (YOLO mode).
- **--approval-mode:** Set the approval mode (default, auto_edit, yolo).
- **--telemetry:** Enable telemetry.
- **--telemetry-target:** Set the telemetry target (local or GCP).
- **-c, --checkpointing:** Enable checkpointing for file edits.
- **--experimental-acp:** Start the agent in ACP mode.
- **-e, --extensions:** Specify extensions.
- **-l, --list-extensions:** List all available extensions.

```
gemini-cli -hUsage: gemini [options] [command]Gemini CLI - Launch an interactive CLI, use -p/--prompt for non-interactive mode
```

Installation on Kali Linux

Install Gemini CLI on Kali Linux by following these steps. Ensure your system runs Kali Linux 2025.3 or newer, as this version includes the required dependencies and support.

Step 1: Update Your System

First, update your system to obtain the latest packages and security updates.

Step 2: Install Dependencies

Gemini CLI requires several dependencies. Install them using the command below.

Step 3: Clone Repository

Clone the Gemini CLI repository from GitHub.

Step 4: Set Up Virtual Environment

Manage Gemini CLI's dependencies efficiently by using a virtual environment.

Configuration and Verification

Step 5: Install Gemini CLI

With the virtual environment active, install Gemini CLI using pip.

```sh
pip install .
```

Step 6: Configure Gemini CLI

Configure Gemini CLI with API keys and essential settings. Create a configuration file in `~/.config/gemini-cli/`.

```sh
mkdir -p ~/.config/gemini-cli/
nano ~/.config/gemini-cli/config.yaml
```

Add the following content to `config.yaml`, replacing placeholders with your API keys and settings:

```yaml
api_key: "your_api_key_here"
api_secret: "your_api_secret_here"
base_url: "https://api.gemini.com"
rate_limit: 120
output_format: "json"
```

Step 7: Verify Installation

Verify Gemini CLI installation and configuration by running:

```sh
gemini-cli --version
```

The command should display Gemini CLI version information, confirming successful installation.

Step 8: Basic Usage Examples

Begin interacting with the Gemini API using Gemini CLI. Here are some basic commands:

1. **Check Account Balances:**

```sh
gemini-cli account balances
```

2. **Place a Limit Order:**

```sh
gemini-cli order new --symbol BTCUSD --amount 0.01 --price 50000 --side buy --type limit
```

3. **Retrieve Market Data:**

```sh
gemini-cli marketdata --symbol BTCUSD
```

We urge you to review our Youtube demonstration videos for example use cases and amplifying information and guides at https://youtube.com/@SecureInSecurity