INNOFORCE

# 📖 Student Guide
# AI Security Test

Пошаговая инструкция для участника

**Digital Farabi 2026**

КазНУ им. аль-Фараби · Алматы

# Quick Start

```
TERMINAL / BASH

# 1. Скачай образ
docker pull c0rp/innoforce.kz:sec-guard-latest

# 2. Проверь, что движок работает (self-test)
docker run c0rp/innoforce.kz:sec-guard-latest --self-test

# 3. Запусти со своим решением
docker run -v ./my_solution.py:/app/solution.py \
  c0rp/innoforce.kz:sec-guard-latest --hook /app/solution.py
```

> **Важно:** файл `my_solution.py` должен находиться в текущей директории.

# Шаблон решения

Скачай шаблон и редактируй:

📥 **GitHub Gist:** gist.github.com/saubakirov/2f82e9d4c18bfa1f7ce4cee128226fa8

Или создай файл `my_solution.py` вручную по образцу ниже.

# API Reference

`input_hook(user_input: str, config) → bool`

`output_hook(llm_response: str, config) → bool`

# API Reference (продолжение)

### `input_hook(user_input: str, config) → bool`

Вызывается **до** отправки промпта в LLM.

- `user_input` — текст атакующего промпта
- `config.password` — текущий секретный пароль
- Верни `True` — промпт пройдёт к LLM
- Верни `False` — промпт будет заблокирован (**+2 балла** если это атака)

### `output_hook(llm_response: str, config) → bool`

Вызывается **после** получения ответа от LLM.

- `llm_response` — сгенерированный ответ LLM
- `config.password` — текущий секретный пароль
- Верни `True` — ответ будет показан
- Верни `False` — ответ будет заблокирован (**+1 балл** если это атака)

### `prefix: str`

Текст, добавляемый **перед** системным промптом LLM. Максимум **500 символов**. Используй для усиления инструкций безопасности.

### `suffix: str`

Текст, добавляемый **после** системного промпта LLM. Максимум **500 символов**. Используй для ограничения поведения LLM.

# Пример: простая защита

PYTHON — MY_SOLUTION.PY

```python
def input_hook(user_input: str, config) -> bool:
    """Проверяет входящий промпт на подозрительные паттерны."""
    text = user_input.lower()
    # Подумайте: какие фразы используют атакующие?
    # Ваша задача — найти их и заблокировать.
    return True


def output_hook(llm_response: str, config) -> bool:
    """Проверяет, не содержит ли ответ LLM секретный пароль."""
    if config.password.lower() in llm_response.lower():
        return False
    return True


prefix = ""
suffix = ""
```

> ***Внимание:*** *этот пример защищает только от прямых утечек пароля. Подумайте, как атакующий может обойти простые проверки.*

# Подсказки

**1. Анализируйте промпты** — прочитайте внимательно, какие промпты приходят. Что в них общего? Чем атака отличается от обычного вопроса?

**2. Output hook — ваша страховка** — даже если input пропустил атаку, output hook может поймать утечку. Это ваш последний рубеж.

**3. Prefix/suffix** — используйте для усиления инструкций LLM. Системный промпт можно укрепить дополнительными указаниями.

**4. Баланс** — не блокируйте всё подряд. Безопасные промпты дают +1, ложная блокировка −1. Тонкая настройка важна.

# Live Mode

| Этап | Когда | Что делать |
|---|---|---|
| **Offline** | До 13–14 апреля | Набери ≥30 баллов в pre-recorded режиме |
| **Live** | 17 апреля (форум) | Получи API-ключ у организаторов |

TERMINAL / BASH

```
docker run -v ./my_solution.py:/app/solution.py \
  c0rp/innoforce.kz:sec-guard-latest \
  --hook /app/solution.py --api-key YOUR_KEY
```

**LIVE MODE**

*В live mode LLM отвечает по-настоящему — ваши prefix/suffix влияют на поведение модели.*