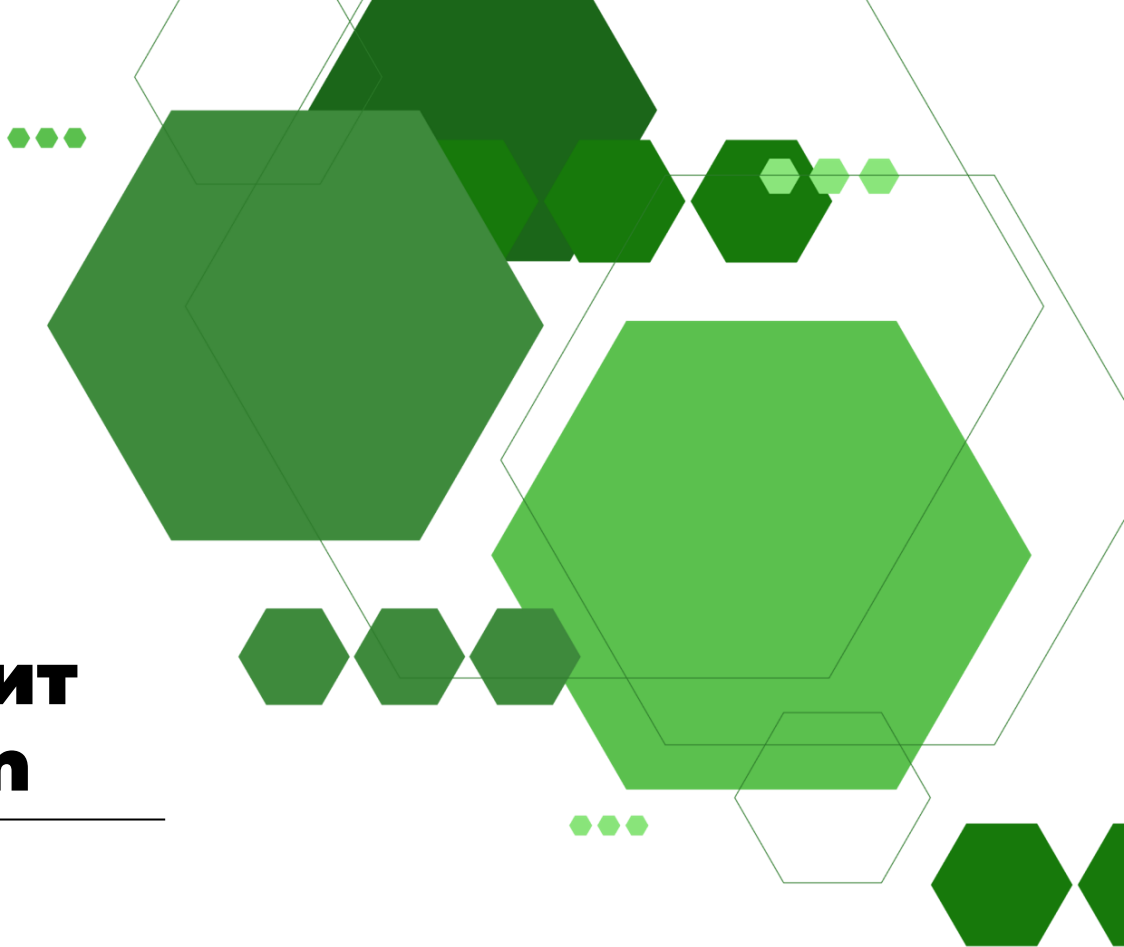


SKILLFACTORY

# Написание утилит для ИБ на Python

---

Контактное занятие



## Обо мне

Аналитик веб-угроз подразделения Threat Intelligence (киберразведка) компании Group-IB

Выпускник МИФИ, направления “Информационно-аналитические системы безопасности”

2 года опыта работы в интеграторе средств защиты информации для объектов критической инфраструктуры



Telegram: @nickotoko

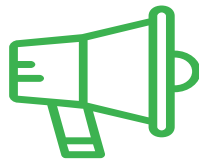
# Правила занятия



**Отложите все  
дела на время  
занятия**



**Фиксируйте  
важную  
информацию  
на вебинаре**



**Задавайте  
вопросы в чат**



**Делитесь  
мнением в  
чате**



**Ура! Вы  
превосходны!**

*Запись семинара будет выложена на платформе “записи встреч”  
[lms.skillfactory.ru](https://lms.skillfactory.ru) в канале в слак*

# План занятия

1. Сканирование сети
2. Application Programming Interface (API)
3. Библиотека requests
4. Command Line Interface (CLI)
5. Реализация фаззера директорий

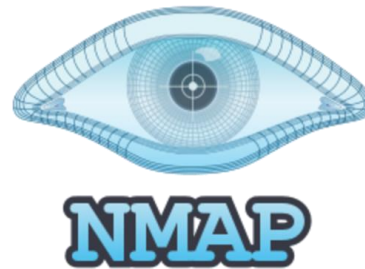


# Сканирование сети

Задачей сканирования сети является выявление хостов в сети, определение запущенных на хостах сетевых сервисов, а также определение возможных уязвимостей данных сервисов.

Выделяют три группы сетевых сканеров:

- 1.Сканеры хостов
- 2.Сканеры портов
- 3.Сканеры уязвимостей



OWASP  
Zed Attack Proxy



# Сканирование сети техникой ping sweep

Ping - это утилита командной строки, которая используется для проверки доступности, целостности и качества соединений в сетях.

Пример команды ping:

```
ping [ключи] <имя сервера или IP-адрес>
```

Утилита ping работает на базе протокола сетевого уровня ICMP



# Application Programming Interface (API)

Интерфейс — это посредник, некоторая прослойка при взаимодействии двух

API (Application Programming Interface, “программный интерфейс приложения”) — это интерфейс для обмена различными данными.

Наиболее популярные технологии для создания API: протокол SOAP и архитектура REST

# Формат передачи данных

Наиболее распространенным форматом передачи данных является JSON (от англ. JavaScript Object Notation).

Второй популярный формат передачи данных это XML (eXtensible Markup Language, «расширяемый язык разметки»)

```
{  
  "firstName": "Иван",  
  "lastName": "Иванов",  
  "address": {  
    "street": "ул. Москворечье, д. 35",  
    "city": "Москва",  
  },  
  "phoneNumbers": [  
    "888 777 66 55",  
    "999 888 77 66"  
  ]  
}
```

```
▼<breakfast_menu>  
  ▼<food>  
    <name>Belgian Waffles</name>  
    <price>$5.95</price>  
    <description>Two of our famous Belgian Waffles with  
    <calories>650</calories>  
  </food>  
  ►<food>  
    ...  
  </food>  
  ►<food>  
    ...  
  </food>  
  ►<food>
```



## HTTP-методы

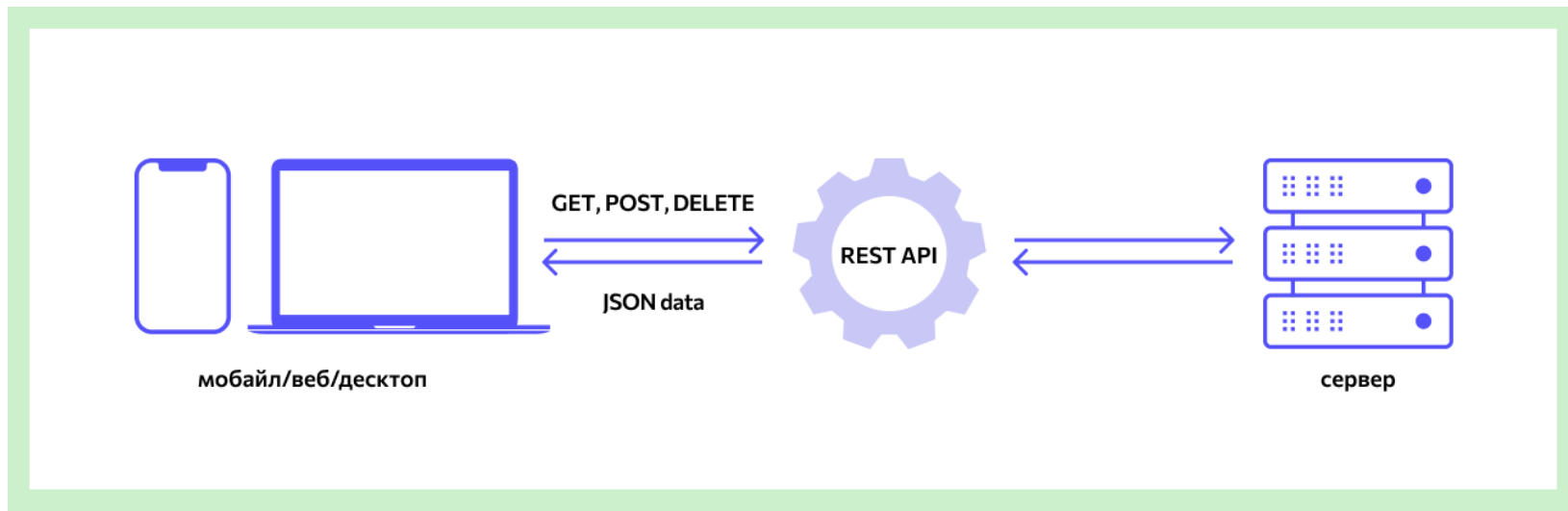
HTTP-метод запроса определяет, что следует сделать:

1. **GET** получает ресурсы;
2. **POST** создаёт ресурс;
3. **PUT** заменяет существующий ресурс целиком;
4. **PATCH** частично изменяет существующий ресурс;
5. **DELETE** удаляет ресурс.

Реже применяют ещё два метода:

1. **HEAD** получает только заголовки ответа
2. **OPTIONS** получает перечень доступных HTTP-методов.

# Application Programming Interface (API)



# Application Programming Interface (API)

Взаимодействовать с серверами в интернете можно не только с использованием браузера.

Браузер используется в качестве клиента для отправки HTTP запросов и получения от серверов HTTP ответов (из содержимого этих ответов в браузере отображается контент, знакомый глазу).

Python позволяет реализовать собственный клиент с использованием библиотеки requests.

## Заголовки запроса:

```
POST /profile.php HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 35
Origin: http://example.com
Connection: keep-alive
Referer: http://example.com/profile.php
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
```

## Тело запроса:

```
username=admin&user_password=123456
```

## Command Line Interface (CLI)

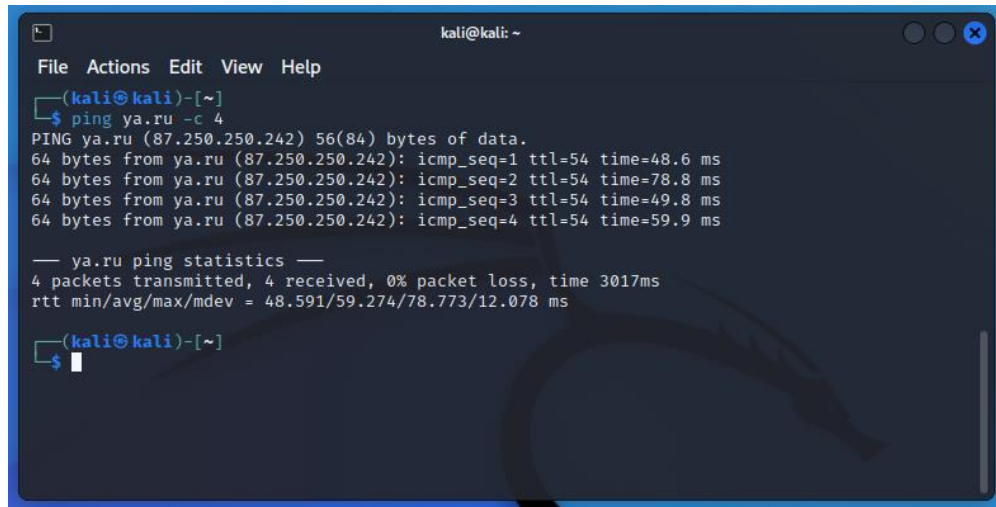
CLI (Command Line Interface, “интерфейс командной строки”) – способ взаимодействия между человеком и компьютером посредством отправки второму текстовых команд.

Умение работать через CLI является важным навыком любого ИБ специалиста.



# Command Line Interface (CLI)

Для того чтобы взаимодействовать с программами на Python через CLI используется библиотека **argparse**.

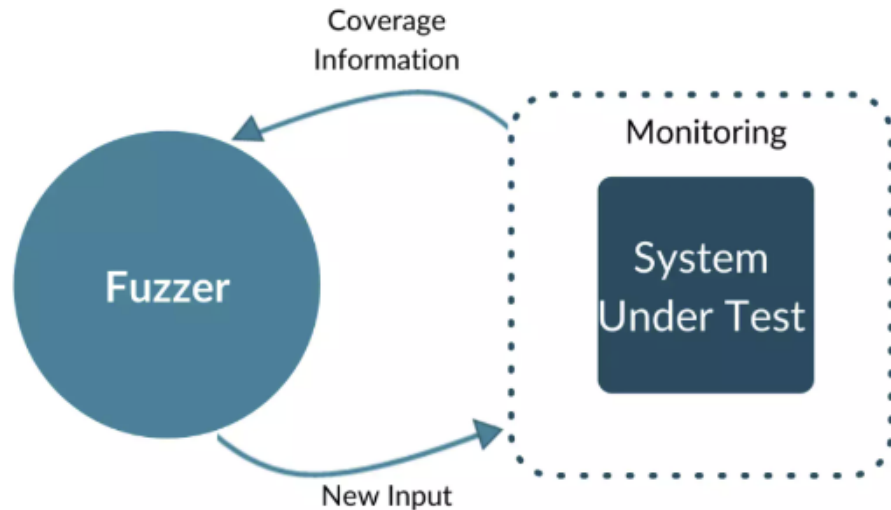


```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ping ya.ru -c 4  
PING ya.ru (87.250.250.242) 56(84) bytes of data.  
64 bytes from ya.ru (87.250.250.242): icmp_seq=1 ttl=54 time=48.6 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=2 ttl=54 time=78.8 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=3 ttl=54 time=49.8 ms  
64 bytes from ya.ru (87.250.250.242): icmp_seq=4 ttl=54 time=59.9 ms  
  
— ya.ru ping statistics —  
4 packets transmitted, 4 received, 0% packet loss, time 3017ms  
rtt min/avg/max/mdev = 48.591/59.274/78.773/12.078 ms  
  
(kali@kali)-[~]  
$
```

# Command Line Interface (CLI)

На основе полученных знаний попробуем реализовать фаззер директорий

Фаззинг — техника тестирования программного обеспечения, часто автоматическая или полуавтоматическая, заключающаяся в передаче приложению на вход неправильных, неожиданных или случайных данных.



# Вопрос-ответ

