Лабораторная работа по проникновению через внешний периметр

Раздел 1: Сбор информации из открытых источников

Задача 1: Поиск поддоменов
Выполним поиск поддоменов различными утилитами.
Начнем с subfinder:

```
subfinder -nW -oI -o cybered.log -d cyber-ed.ru
```

Применим GAU:

```
GO111MODULE=on go get -u -v github.com/lc/gau
# Extract subdomains from output
gau -subs cyber-ed.ru | cut -d / -f 3 | sort -u
```

И amass:

```
amass -active -brute -o hosts.txt -d cyber-ed.ru
```

Задача 2: анализ логов certificate transparency
Проведем сбор поддоменов по логам certificate transparency:

```
$ sudo apt-get install postgresql-client-12
$ wget https://raw.githubusercontent.com/hannob/tlshelpers/master/
getsubdomain
$ bash getsubdomain cyber-ed.ru
```

Альтернативный вариант - web-интерфейс: https://crt.sh/

Раздел 2: Активная разведка DNS

Задача 1: Reverse DNS lookup
Выполнение запросов reverse DNS:

```
dig axfr @nsztm1.digi.ninja zonetransfer.me
```

Задача 2: трансфер DNS-зоны
Выполнение трансфера зоны:

```
dig axfr @nsztm1.digi.ninja zonetransfer.me
```

Задача 3: перебор DNS-имен по словарю
Используем altdns:

```
# Altdns is a DNS recon tool that allows for the discovery of subdomains
that conform to patterns. Altdns takes in words that could be present in
subdomains under a domain (such as test, dev, staging) as well as takes in
a list of subdomains that you know of.
# https://github.com/infosec-au/altdns
pip install py-altdns
# Basic usage
altdns -i known-subdomains.txt -o raw-data_output -w words.txt -r -s
results_output.txt
```

Ускорим перебор за счет использования massdns.
Установим dnsgen для генерации словаря доменов.

```
pip3 install dnsgen
```

Соберем massdns из исходного кода:

```
git clone https://github.com/blechschmidt/massdns.git
cd massdns
make
```

Подготовим список резолверов:

```
echo -en '8.8.8.8\n8.8.4.4\n1.1.1.1\n' > resolvers.txt
```

Начнем перебор доменов в комбинации с dnsgen:

```
cat domains.txt | dnsgen -w words.txt -f - | massdns -r resolvers.txt -t A
-o S -w massdns.out
```

Раздел 3: Разведка веб-приложений

Задача 1: обнаружение endpoint-ов
Выполним обнаружение живых endpoint-ов с помощью httprobe:

```
cat domains.txt | httprobe -p http:8080 -p https:8443 -c 16
```

Задача 2: поиск виртуальных хостов
Для работы с виртуальными хостами можем использовать curl:

```
curl -H "Host: example.com" http://localhost/
```

Для перебора можем использовать утилиты gobuster, wfuzz, vhostbrute или VHostScan:

```
gobuster vhost -u https://site.com -t 50 -w subdomains.txt
wfuzz -c -w subdomains.txt --hc 400,404,403 -H "Host: FUZZ.example.com" -u
http://example.com -t 100
vhostbrute.py --url="example.com" --remoteip="10.1.1.15" --
base="www.example.com" --vhosts="vhosts_full.list"
VHostScan -t example.com
```

Задача 3: определение web-технологий
Используем whatweb для определения технологий и ПО, используемого веб-приложениями:

```
whatweb -v -a 3 --log-brief=whatweb.log -i domains.txt
```

Задача 4: визуальный анализ
Проведем визуальный анализ, создав скриншоты с помощью утилиты aquatone:

```
cat domains.txt | aquatone
```

Задача 5: сканирование nuclei
Используем nuclei для сканирования хоста на уязвимости:

```
nuclei -l domains.txt
```

Задача 6: сканирование CMS
Просканируем CMS WordPress с помощью специализированного сканера WPScan:

```
wpscan -url http://target.domain
```

Задача 7: перебор файлов и директорий
Ведение перебора файлов:

```
gobuster dir -t 128 -u http://target.domain/ -w /usr/share/seclists/
Discovery/Web-Content/raft-large-words.txt -x php,html,aspx
```

Задача 8: crawling и работа с параметрами
Выполним crawling сайта с gospider:

```
gospider -s "http://target.com/" -o output -c 10 -d 1
```

Используем arjun для поиска параметров:

```
$ python3 arjun.py -u https://api.example.com/endpoint --get
```