

Пассивная разведка

Введение

Пассивная разведка — способ получить информацию о целевой системе, действуя легитимно (например, просмотр веб-страницы или покупка в онлайн-магазине) или вовсе без взаимодействия с ней. Опираясь таким образом, вы не рискуете обнаружить себя как «злоумышленника», однако платой за бесшумность будет время, надёжность и актуальность результатов. Такому сбору информации подвержены все компании будь это малый бизнес или крупные государственные предприятия.

Технические методы, которые используются при пассивной разведке:

1. **Google** постоянно индексирует (сканирует своим ботом) веб-страницы всего интернета и записывает к себе в базу.
2. **Анализ DNS-записей.** Хорошая [демонстрация](#) как работает *DNS*. Поиск и анализ *DNS*-записей – задача, которая ставится практически при каждом пентесте. Это очень важный этап, позволяющий собрать карту доменов и *IP*-адресов. *DNS* – один из самых атакуемых сетевых протоколов.
3. **Censys, shodan** и пр. ресурсы, которые сканируют сеть. С помощью этих ресурсов можно найти сетевые адреса, доменные имена, баннеры сетевых служб, используемых на внешнем периметре сети. По баннерам часто можно определить версию используемого ПО, а также некоторые уязвимости этих служб.
4. **Анализ трафика Wireshark, tcpdump** и прочие инструменты позволяющие анализировать сетевой трафик. Анализируя трафик от сервера, можно определить ПО и его компоненты, версию. В случае с веб-сервером, можно проанализировать скрипты и *HTML*-код для того, чтобы понять как работает веб-приложение, куда обращается (могут быть внутренние или внешние ресурсы).
5. **Поиск утечек по учётным записям и почтовым адресам.** Сотрудники могут использовать одинаковый пароль для личных и корпоративных учётных записей, это может сыграть на руку и расширить диапазон для поиска почтовых ящиков и учётных записей. Даже если пароль не актуален, его можно использовать как паттерн для дальнейшего подбора. Сам факт утечки можно проверить [на этом сервисе](#) (один из вариантов).
6. **Поиск файлов, связанных с целью на virustotal.** *Virustotal* – ресурс, позволяющий проверить файл на вредоносность. Все файлы, загруженные на проверку остаются в системе и их может найти и скачать любой желающий. Часто можно найти конфиденциальные документы по имени. Для загрузки файлов требуется регистрация и подписка.

Google Dorks

<https://www.exploit-db.com/google-hacking-database>

Комбинируя указанные операторы между собой можно добиться интересных результатов.

Чтобы нагуглить такую информацию необходимо использовать операторы поиска или так называемые *Google Dorks*. Самые основные операторы:

- **Site:** поиск на конкретном сайте (пример — *site:example.com*).
- **inurl:** поиск слова в *URL* (*inurl:config.txt*).
- **intext:** поиск в ответе сервера (*intext:* «слово или словосочетание»).
- **filetype:** поиск по типам файлов (*filetype: pdf*). Оператор *filetype* поддерживает поиск по следующим форматам: *Adobe Flash (.swf)*, *Adobe Portable Document Format (.pdf)*, *Adobe PostScript (.ps)*, *Autodesk Design Web Format (.dwt)*, *Google Earth (.kml, .kmz)*, *GPS eXchange Format (.gpx)*, *Hancom Hanword (.hwp)*, *HTML (.htm, .html, other file extensions)*, *Microsoft Excel (.xls, .xlsx)*, *Microsoft PowerPoint (.ppt, .pptx)*, *Microsoft Word (.doc, .docx)*, *OpenOffice presentation (.odp)*, *OpenOffice spreadsheet (.ods)*, *OpenOffice text (.odt)*, *Rich Text Format (.rtf)*, *Scalable Vector Graphics (.svg)*, *TeX/LaTeX (.tex)*, *Text (.txt, .text, other file extensions)*, *Basic source code (.bas)*, *C/C++ source code (.c, .cc, .cpp, .cxx, .h, .hpp)*, *C# source code (.cs)*, *Java source code (.java)*, *Perl source code (.pl)*, *Python source code (.py)*, *Wireless Markup Language (.wml, .wap)*, *XML (.xml)*.

Полный список

Оператор	Описание	Синтаксис	Пример
()	Группировка нескольких терминов или операторов. Позволяет строить сложные выражения	(<термин> или <оператор>)	`inurl:(html
*	Подстановочный знак. Соответствует любому слову	<текст> * <текст>	Как * компьютер
" "	Точное совпадение фразы (без учёта регистра)	"<ключевые слова>"	"Google"
m..n / m...n	Поиск чисел в диапазоне (n должно быть больше m)	<число>..<число>	1..100
-	Исключает документы, соответствующие оператору (аналог NOT)	-<оператор>	-site:youtube.com
+	Включает документы, соответствующие оператору	+<оператор>	+site:youtube.com
\	Логическое ИЛИ. Достаточно совпадения одного из операторов	<оператор> \ <оператор>	"Google" \ "Yahoo"
~	Поиск синонимов слова (не поддерживается Google)	~<слово>	~книга
@	Поиск только на указанной соцсети (лучше использовать site:)	@<соцсеть>	@instagram
after:	Поиск документов, опубликованных после указанной даты	after:<год(-месяц-день)>	after:2020-06-03
allintitle:	Поиск по нескольким ключевым словам в заголовке (через пробел)	allintitle:<ключевые слова>	allintitle:собака кошка
allinurl:	Поиск по нескольким ключевым словам в URL (через пробел)	allinurl:<ключевые слова>	allinurl:поиск com
allintext:	Поиск по нескольким ключевым словам в тексте документа (через пробел)	allintext:<ключевые слова>	allintext:математика наука
AROUND()	Поиск документов, где первое слово находится в пределах n слов от второго	<слово1> AROUND(<n>) <слово2>	Google AROUND(10) хороший
author:	Поиск статей указанного автора (если применимо)	author:<имя>	author:Иван Петров
before:	Поиск документов, опубликованных до указанной даты	before:<год(-месяц-день)>	before:2020-06-03
cache:	Поиск в кэшированной версии сайта (использует кэш Google)	cache:<домен>	cache:google.com
contains:	Поиск документов, содержащих ссылки на указанный тип файла (не поддерживается Google)	contains:<тип_файла>	contains:pdf
date:	Поиск документов, опубликованных за последние n месяцев (не поддерживается Google)	date:<число>	date:3
define:	Поиск определения слова	define:<слово>	define:смешной
ext:	Поиск файлов определённого типа	ext:<тип_файла>	ext:pdf
filetype:	Аналог ext:	filetype:<тип_файла>	filetype:pdf

Оператор	Описание	Синтаксис	Пример
<code>inanchor:</code>	Поиск ключевого слова в анкерах (якорных ссылках)	<code>inanchor:</code> <code><ключевое_слово></code>	<code>inanchor:безопасность</code>
<code>index of:</code>	Поиск документов с прямыми загрузками	<code>index of:<термин></code>	<code>index of:mp4 видео</code>
<code>info:</code>	Поиск информации о сайте	<code>info:<домен></code>	<code>info:google.com</code>
<code>intext:</code>	Ключевое слово должно быть в тексте документа	<code>intext:</code> <code><ключевое_слово></code>	<code>intext:новости</code>
<code>intitle:</code>	Ключевое слово должно быть в заголовке документа	<code>intitle:</code> <code><ключевое_слово></code>	<code>intitle:деньги</code>
<code>inurl:</code>	Ключевое слово должно быть в URL документа	<code>inurl:</code> <code><ключевое_слово></code>	<code>inurl:таблица</code>
<code>link: / links:</code>	Поиск документов, ссылающихся на указанный ключевой запрос (или сайт)	<code>link:</code> <code><ключевое_слово></code>	<code>link:google</code>
<code>location:</code>	Показывает документы, связанные с указанным местоположением	<code>location:<место></code>	<code>location:Россия</code>
<code>numrange:</code>	Аналог <code>m..n</code> (поиск чисел в диапазоне)	<code>numrange:<число>-<число></code>	<code>numrange:1-100</code>
OR	Аналог <code>\ </code> (логическое ИЛИ)	<code><оператор> OR <оператор></code>	<code>"Google" OR "Yahoo"</code>
<code>phonebook:</code>	Поиск телефонных номеров, связанных с указанным именем	<code>phonebook:<имя></code>	<code>phonebook:"Иван Иванов"</code>
<code>related:</code>	Поиск сайтов, похожих на указанный	<code>related:<домен></code>	<code>related:google.com</code>
<code>safesearch:</code>	Исключает взрослый контент (например, порнографию)	<code>safesearch:</code> <code><ключевое_слово></code>	<code>safesearch:секс</code>
<code>source:</code>	Поиск на указанном новостном сайте (лучше использовать <code>site:</code>)	<code>source:</code> <code><новостной_сайт></code>	<code>source:bbc</code>
<code>site:</code>	Поиск только на указанном сайте (можно использовать домен верхнего уровня, например <code>.ru</code>)	<code>site:<домен></code>	<code>site:google.com</code>
<code>stock:</code>	Поиск информации об указанной акции	<code>stock:<акция></code>	<code>stock:Газпром</code>
<code>weather:</code>	Поиск информации о погоде в указанном месте	<code>weather:<место></code>	<code>weather:Москва</code>

Анализ DNS-записей

Из такого поиска может получиться найти администратора домена или контактное лицо, ответственное за этот ресурс – можно использовать полученные данные для дальнейшей социальной инженерии. Может быть, получится найти какие-либо поддомены.

Общедоступные ресурсы записей DNS:

<https://dnsdumpster.com/>

<https://whois.ru/>

`https://whois.ru/URL`

Проверка WAF

`wafw00f`

kali linux

Recon DNS

```
dnsrecon -d site.com
```

```
host {SITE>COM}
```

```
whatweb site.com
```

Либо можно поспрашивать у *DNS*-серверов самостоятельно используя следующие инструменты:

- [Photon](#) — многофункциональный инструмент для проведения рекона. Изучение содержимого целевых ресурсов происходит из веб-архива (т. е. без взаимодействия с целевой инфраструктурой).

KALI LINUX

```
sudo apt update
sudo apt install photon
```

Поиск информации о yandex.ru

```
photon -u "https://yandex.ru" -l 3 -t 2 -v --keys --wayback -o /root/photon
```

```
-rw-r--r-- 1 root root 9.0M Nov 12 11:26 custom.txt
-rw-r--r-- 1 root root 19K Nov 12 11:26 endpoints.txt
-rw-r--r-- 1 root root 25K Nov 12 11:26 external.txt
-rw-r--r-- 1 root root 211 Nov 12 11:26 files.txt
-rw-r--r-- 1 root root 40K Nov 12 11:26 fuzzable.txt
-rw-r--r-- 1 root root 21K Nov 12 11:26 intel.txt
-rw-r--r-- 1 root root 72K Nov 12 11:26 internal.txt
-rw-r--r-- 1 root root 5.0K Nov 12 11:26 robots.txt
-rw-r--r-- 1 root root 680 Nov 12 11:26 scripts.txt
```

- [Sublist3r](#) — инструмент для поиска поддоменов. Поиск поддоменов производится с помощью известных поисковых систем (в т. ч. *Google*) и несколько дополнительных ресурсов.

```
sudo apt update
sudo apt install photon
```

```
sublist3r -d "yandex.ru" -v -t 2
```

- [theHarvester](#) — так же, как и *Photon*, многофункциональный фреймворк для проведения пассивной разведки.

```
theHarvester -d yandex.ru -b
baidu,bing,bingapi,bufferoverun,certspotter,crtsh,dnsdumpster,duckduckgo,exalead,github-
code,google,hackertarget,hunter,intelx,linkedin,linkedin_links,netcraft,otx,pentesttools,projectdi
scovery,qwant,rapiddns,securityTrails,spyse,sublist3r,threatcrowd,threatminer,trello,twitter,urlsc
an,virustotal,yahoo
```

Для некоторых поисковиков может потребоваться *API*-ключ.

Поиск по «онлайн»-сканерам.

- [Censys](#) — сервис, позволяющий просматривать результаты сканирования доменных имен и *IP*-адресов. Условно бесплатный (ограничение запросов на *IP*-адрес). Имеет довольно гибкий синтаксис запросов, который хорошо [задокументирован](#). Этот сервис, как и следующий (*shodan*), использует различные *OSINT* (*open source intelligence*) *framework*, например тот же *theHarvester*.
- [Shodan](#) — аналог *Censys*, но с более простым интерфейсом и языком запросов.
- [Zoomeye](#) — когда *Censys* или *Shodan* не помогли, и ты знаешь китайский :).
- [URLScan](#) — онлайн «песочница» для веба. Функционал довольно простой: этот сервис любезно сходит на указанный веб-сайт и соберёт всю важную информацию с него и сделает скриншоты.

Поиск по утечкам паролей.

Как и указывалось ранее, существуют сервисы проверки учётных записей на предмет утечки паролей, самый крупный — [Have I Been Pwned](#), однако сервис не укажет пароль в явном виде, но укажет, в каких утечках засветился тот или иной аккаунт. Чтобы найти пароль, следует самостоятельно найти, скачать и проанализировать утечку.

BreachDirectory Бесплатный поиск по email и хешам паролей.

<https://breachdirectory.org>

<https://intelx.io/>

<https://haveibeenpwned.com/>

Браузерные расширения

BuiltWith

Wapolizer

Копия WEB-сайта

<https://httrack.com>

```
sudo apt install webhttrack
```

==Активная разведка==

Введение

Активный сбор информации — процесс получения данных о исследуемой инфраструктуре путём прямого взаимодействия с ней. Примерами активной разведки являются:

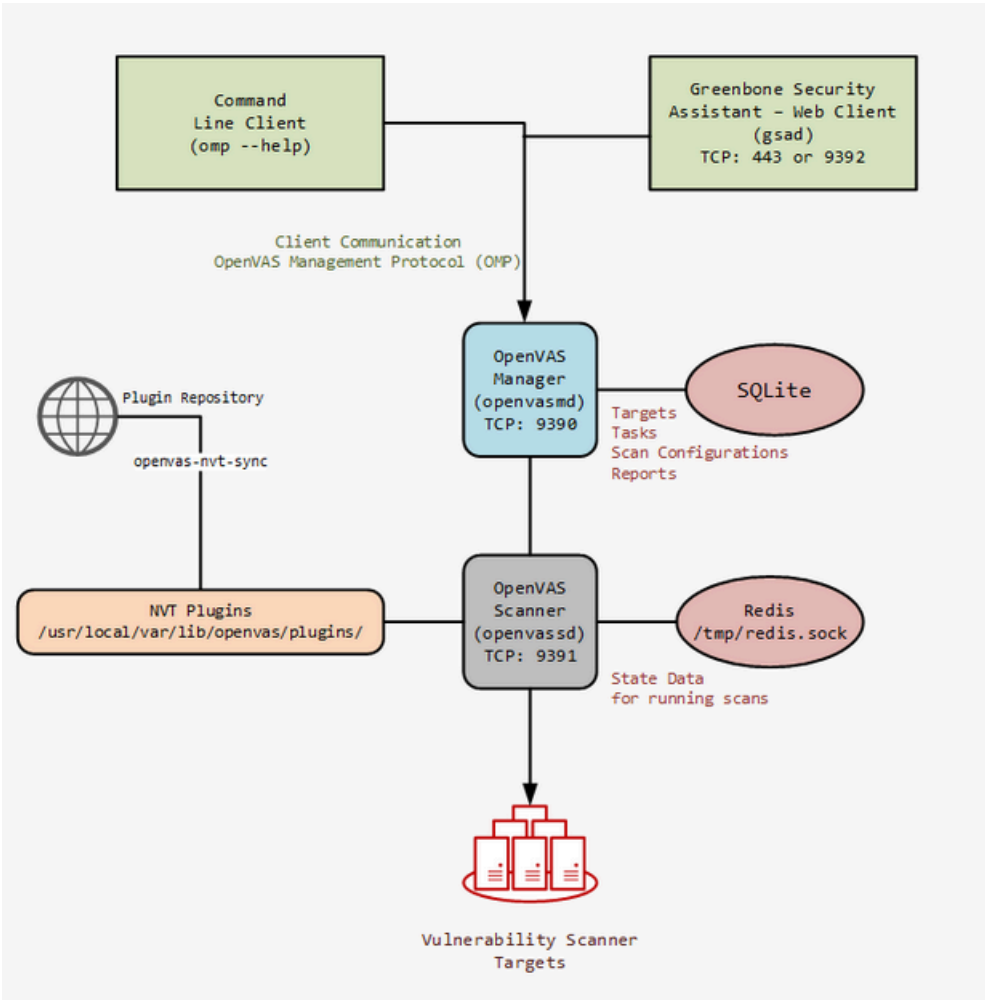
1. Сканирование сети.
 2. Поиск уязвимостей с помощью различных инструментов.
 3. Получение и изучение баннеров периметровых сервисов.
- kali-linux

```
dnsenum
```

Инструменты активной разведки

OpenVAS

Инструмент, который управляется с помощью веб-интерфейса. Можно установить как [контейнерную версию](#) (самый легкий путь), так и полностью на [Kali Linux](#).



OpenVAS при сканировании уязвимостей опирается на NVT (*network vulnerability test*) — специальные тесты на те или иные уязвимости, которых даже из коробки больше 45 тысяч, что ощутимо больше, чем lua-скриптов в Nmap. NVT тесты разделяются по «фамилиям» или типам (например Cisco, Windows Bulletins и пр.), и всего их из коробки 62.

DashboardScansAssetsSecInfoConfigurationExtrasAdministrationHelp

Scan Config: Full and very deep ultimate

ID: 74db13d6-7489-11df-91b9-002264764cea
Created: Tue Aug 21 18:53:13 2018
Modified: Tue Aug 21 18:53:13 2018

Comment: Most NVT's including those that can stop services/hosts; don't trust previously collected information; slow.

Network Vulnerability Test Families (62)

Family	NVTs selected	Trend
AIX Local Security Checks	1 of 1	↗
Amazon Linux Local Security Checks	748 of 748	↗
Brute force attacks	9 of 9	↗
Buffer overflow	562 of 562	↗
CISCO	647 of 647	↗
CentOS Local Security Checks	2528 of 2528	↗
Citrix XenServer Local Security Checks	30 of 30	↗
Compliance	7 of 7	↗
Databases	556 of 556	↗
Debian Local Security Checks	3101 of 3101	↗
Default Accounts	245 of 245	↗
Denial of Service	1384 of 1384	↗
F5 Local Security Checks	125 of 125	↗

При нахождении веб-сервисов будут дополнительно (в зависимости от профиля сканирования) запущены

утилиты *Nikto*, *DIRB*, *Wapiti*, и проанализирован их вывод (если анализ в автоматическом режиме не удался, в результатах сканирования показывается вывод работы утилит).

AppsKali LinuxKali TrainingKali ToolsKali DocsKali ForumsNetHunterOffensive SecurityExploit-DB

DashboardScansAssetsSecInfoConfigurationExtrasAdministrationHelp

ReportingUnknown NASL ServicesService Banner



19

012345678910

1 - 10 of 20

Vulnerability	Severity	QoD	Host	Location	Created
CPE Inventory	0.0 (Log)	80%		general/CPE-T	Thu Nov 25 18:28:10 2021
Nikto (NASL wrapper)	0.0 (Log)	98%		80/tcp	Thu Nov 25 18:23:14 2021
DIRB (NASL wrapper)	0.0 (Log)	98%		80/tcp	Thu Nov 25 18:22:15 2021
wapiti (NASL wrapper)	0.0 (Log)	98%		80/tcp	Thu Nov 25 18:20:39 2021
Unknown OS and Service Banner Reporting	0.0 (Log)	80%		general/tcp	Thu Nov 25 18:20:22 2021
Unknown OS and Service Banner Reporting	0.0 (Log)	80%		53/tcp	Thu Nov 25 18:20:22 2021
Unknown OS and Service Banner Reporting	0.0 (Log)	80%		443/tcp	Thu Nov 25 18:20:22 2021
Telnet Unencrypted Cleartext Login	4.8 (Medium)	70%		23/tcp	Thu Nov 25 18:16:15 2021
Traceroute	0.0 (Log)	80%		general/tcp	Thu Nov 25 18:15:52 2021
CGI Scanning Consolidation	0.0 (Log)	80%		80/tcp	Thu Nov 25 18:15:40 2021

Apply to page content

Vulnerability	Severity	QoD	Host	Location	Actions
Nikto (NASL wrapper)	0.0 (Log)	98%		80/tcp	 

Summary

This plugin uses nikto to find weak CGI scripts and other known issues regarding web server security. See the preferences section for configuration options.

Note: The plugin needs the 'nikto' or 'nikto.pl' binary found within the PATH of the user running the scanner and needs to be executable for this user. The existence of this binary is checked and reported separately within 'Availability of scanner helper tools' (OID: 1.3.6.1.4.1.25623.1.0.810000).

Vulnerability Detection Result

Here is the Nikto report:
- Nikto v2.1.6

































+ Target IP:
+ Target Hostname:
+ Target Port: 80
+ Virtual Host:
+ Start Time: 2021-11-25 18:21:49 (GMT0)

+ Server: Web server
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ All CGI directories 'found', use '-C none' to test none
+ 26440 requests: 0 error(s) and 3 item(s) reported on remote host
+ End Time: 2021-11-25 18:23:13 (GMT0) (84 seconds)

+ 1 host(s) tested

Пример вывода работы утилиты *Nikto* в интерфейсе *OpenVAS*

Из коробки 9 версия *OpenVAS* имеет 8 профилей сканирования:

Discovery (Network Discovery scan configuration.)	20	2723	   
empty (Empty and static configuration template.)	0	0	   
Full and fast (Most NVT's; optimized by using previously collected information.)	62	49767	   
Full and fast ultimate (Most NVT's including those that can stop services/hosts; optimized by using previously collected information.)	62	49767	   
Full and very deep (Most NVT's; don't trust previously collected information; slow.)	62	49767	   
Full and very deep ultimate (Most NVT's including those that can stop services/hosts; don't trust previously collected information; slow.)	62	49767	   
Host Discovery (Network Host Discovery scan configuration.)	2	2	   
System Discovery (Network System Discovery scan configuration.)	6	29	   

Nikto

[nikto](#) — это легковесный быстрый сканер веб-ресурсов на предмет мисконфигов и уязвимостей. По умолчанию идёт из коробки в *Kali*-дистрибутиве.

Основные функции сканера:

1. Поиск мисконфигов на веб-серверах.
2. Поиск файлов и директорий по умолчанию (например */cgi-bin/*).
3. Поиск небезопасных файлов на веб-сервере (например */etc/passwd*).
4. Выявление устаревшего ПО.
5. Анализ и подсветка вхождений, требующих ручной проверки (например анализ содержимого *robots.txt*).

Инструмент очень удобный, однако следует помнить о том, что в случае противодействия синей команды, необходимо аккуратно использовать сканер: как минимум, следует задавать *user-agent* (по умолчанию в него проставляется версия *nikto*, на что давно написаны детекты, и это выдаст вас при первом же сканировании) и дополнительно использовать параметр *evasion*. Подробности использования утилиты следует искать в `man nikto`.

To scan a particular host

```
nikto -host [host IP/name]
```

To scan a host on multiple ports (default = 80)

```
nikto -host [host IP/name] -port [port number 1], [port number 2], [port number 3]
```

To scan a host and output fingerprinted information to a file

```
nikto -host [host IP/name] -output [output_file]
```

To use a proxy while scanning a host

```
nikto -host [host IP/name] -useproxy [proxy address]
```

UDP Protocol Scanner

Следующий рассматриваемый инструмент — [udp-proto-scanner](#). При анализе сетевого периметра обычно большее внимание уделяется *TCP*-сервисам, нежели *UDP*. Тем не менее, такие сервисы не следует упускать из виду, и для этого можно использовать указанный инструмент. Основной функцией является определение *UDP* сервиса на *IP*-адресе, причём рекомендуется использовать такой сканер на большие диапазоны сетей (больше чем /24) ввиду особенностей сканирования.

```
git clone https://github.com/CiscoCXSecurity/udp-proto-scanner.git
cd udp-proto-scanner
perl udp-proto-scanner.pl
```



```
(root@kali)~[~]
# git clone https://github.com/CiscoCXSecurity/udp-proto-scanner.git
Cloning into 'udp-proto-scanner'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 32 (delta 0), reused 3 (delta 0), pack-reused 29
Receiving objects: 100% (32/32), 21.59 KiB | 92.00 KiB/s, done.
Resolving deltas: 100% (12/12), done.

(root@kali)~[~]
# cd udp-proto-scanner

(root@kali)~/udp-proto-scanner[~]
# ls
CHANGELOG  COPYING.GPL  COPYING.UDP-PROTO-SCANNER  INSTALL  README.md  udp-proto-scanner.conf  udp-proto-scanner.pl

(root@kali)~/udp-proto-scanner[~]
# perl udp-proto-scanner.pl
ERROR: Supply some hosts to scan.

Usage: udp-proto-scanner.pl [options] [ -p probe_name ] -f ipsfile
      udp-proto-scanner.pl [options] [ -p probe_name ] 10.0.0.0/16 172.16.16.1 192.168.0.1

Options are:
  --file file           File of ips
  --probe_name          Name of probe or 'all' (default: all probes)
  --list_probes         List all available probe name then exit
  --bandwidth n         Bandwidth to use in bits/sec. Default 250k
  --configfile file     Config file to use. Default ./udp-proto-scanner.conf
                        or /etc/udp-proto-scan.conf
  --retries n           No of packets to sent to each host. Default 3
  --help               This message
```