Computer & Information Security

PRACTICAL SESSION NO. 1

INTRO & INFORMATION GATHERING

Agenda

- ❑ Welcome, Assignments and Labs requirements
- ❑Information Gathering

General Information

- ❑ Course team
  - Lectures
    - Prof. Yuval Elovici - elovici@bgu.ac.il
    - Dr. Asaf Shabtai – shabtaia@bgu.ac.il
    - Dr. Mordehai Guri – gurim@post.bgu.ac.il
  - Teaching Assistants
    - Ron Bitton – ronbit@post.bgu.ac.il
    - Ben Nassi – nassib@post.bgu.ac.il
    - Aviad Elishar aviade@post.bgu.ac.il
    - Noam moscovich noammosc@post.bgu.ac.il
    - Assignments: Vitaly Dyadyuk- vitalyd@post.bgu.ac.il
- ❑ Course’s mail - itns.ise@gmail.com
- ❑ Course’s site - Moodle

General Information

- Any question with regards to assignments, lectures, practical-sessions or labs should be directed to the appropriate forum in the course web-site.
- Administrative questions should be directed to the course email.
- For the very urgent topics, or if you haven't received an answer from the course email within three days - you can directly approach to Ron Bitton (ronbit@post.bgu.ac.il).

Practical Sessions and Labs

- The course include both theoretical sessions (5) and labs (8)
- Each student require to submit a solution to **<u>7</u>** out of 8 labs.

#	Lab/Theoretical-session	Topic	Date	Responsible TA
1	Lab (1)	Information Gathering	Feb 24th weak	Ron Bitton
2	Theoretical-session (1)	Introduction to cryptography	Mar 3th weak	Ben Nassi
3	Theoretical-session (2)	Symmetric key cryptography	Mar 10th weak	Ben Nassi
4	Theoretical-session (3)	Public key infrastructure	Mar 24th weak	Ben Nassi
5	Theoretical-session (4)	Authentication Protocols	Mar 31th weak	Ron Bitton
6	Lab (2)	Introduction to Linux access control	Apr 7th weak	Ron Bitton
7	Lab (3)	POSIX interface and social engineering attacks	Apr 28th weak	Ron Bitton
8	Theoretical-session (5)	Firewalls	May 12th weak	Aviad Elishar
9	Lab (4)	Understanding buffer overflow attacks	May 19th weak	Noam Moskovich
10	Lab (5)	Web application security using OWASP(1)	May 26th weak	Noam Moskovich
11	Lab (6)	Web application security using OWASP (2)	Jun 2th weak	Noam Moskovich
12	Lab (7)	Malware analysis using Cuckoo sandbox	Jun 10th weak	Aviad Elishar
13	Lab (8)	Vulnerability assessment and penetration testing using OpenVAS and Metasploit	Jun 17th weak	Aviad Elishar

Assignments

- The course include **4** assignments (submitted in **pairs**)

#	Topic	Dates	Responsible TA	Responsible Lecture
1	Cryptography and Wireshark	17/03/2019 – 04/04/2019 (23:59)	Ben Nassi	Asaf Shabtai
2	Authentication protocols and Linux access control	28/04/2019 – 16/05/2019 (23:59)	Ron Bitton	Asaf Shabtai
3	Web application security, network attacks and BOF	19/05/2019 – 06/06/2019 (23:59)	Noam Moscovich	Mordechai Guri
4	Final assignment (all course topics)	09/06/2019 – 21/06/2019 (23:59)	Aviad Elishar	Yuval Elovici

- Delay will not be allowed

Information Gathering

Information Gathering

Definition:

- “the process of collecting information about something”

More specifically:

- **Military:** the operation of gathering information about an enemy
- **Information Security:** collecting as much information as possible about a target

Information Gathering

Information we can collect:

- ❑ **Network Information:** Port Scanning, Topology, Firewalls
- ❑ **Devices information:** Hardware, Operating System, Apps
- ❑ **Users Activity:** Account and browsing Information
- ❑ **Property:** Ownership of IP & Domains, Docs

And many other interesting things…

Information Gathering

Main Purpose:

❑Understand **How** to Attack/Defend

❑ Understand **Where** to Attack/Defend

❑ Understand **When** to Attack/Defend

❑ Understand **Who** to Attack/Defend

❑ Understand How/Where/When/Who **NOT** to Attack

This information is highly valuable!

How To Gather Information?

- **Network Scans:** extract information from network structure and components.
- **Eavesdropping:** listening to the private conversation (or communication of others without their consent
- **Dumpster Diving**: looking for treasure in someone else’s trash.
- **Social Engineering:** manipulating people to perform actions.

KALI LINUX

- Free of charge customizable open source **Debian-based** Linux distribution aimed at advanced penetration testing (PT) and security auditing.
- Contains over **600** verified tools which are geared towards various information security tasks
- Widely used by security researchers for PT, computer forensics and reverse engineering.
- Developed, funded and maintained by Offensive Security Ltd

Few of the powerful tools Kali provides

- ❑ Wireshark (packet analyzer)
- ❑ nmap (port scanner)
- ❑ URLCrazy (domain similarity tester)
- ❑ Firewalk (L4 determination)
- ❑ Parsero (Robot unauthorized scanner)
- ❑ Theharvaster (Entities gathering)

Wireshark

- Network sniffing tool
- Live capture and offline analysis
- Deep inspection of hundreds of protocols
- Network professionals, security experts, developers, and educators around the world use it regularly
- Freely available as open source

<u>The first assignment is consisted of practical exercises in Wireshark</u>

Nmap (**N**etwork **map**per)

Features

- Powerful and popular network scanning tool.
- Can scan huge networks with 100,000s of machines.
- Can detect open ports, listening network services and OS version.

Basic scanning

- The simple command **nmap <target>** scans 1,000 TCP ports on the host (<target>).
  - nmap 192.168.1.1 – scan a single host
  - nmap 192.168.1.1-10 – scan a range of hosts
  - nmap 192.168.1.1/24 – scan a subnet

Nmap (**N**etwork **map**per)

Host Discovery

❑ **PING Scan (-sP/-sn):**

- One of the very first steps in any network reconnaissance mission is to reduce a (sometimes huge) set of IP ranges into a list of active or interesting hosts.
- Scanning every port of every single IP address is slow (and usually unnecessary).
- Ping scan option tells Nmap not to do a port scan after host discovery, and only print out the available hosts that responded to the host discovery probes.
- The default host discovery (a ping scan) for a privilege user consists of actions:
  (1) ICMP echo request.
  (2) TCP SYN to port 443.
  (3) TCP ACK to port 80.
  (4) ICMP timestamp request.

(*) For machines on a local ethernet network, ARP scanning will be performed.

- When executed by an unprivileged user, only SYN packets are sent (i.e., using the connect system call – will be further explained later)

Nmap (Network mapper)

Port Scanning Basics

❑ Nmap is divides ports into six states as follows:

1. **open:** An application is actively accepting TCP connections or UDP datagrams on this port.
2. **closed:** A closed port is accessible (it receives and responds to Nmap probe packets), but there is no application listening on it.
3. **filtered:** Nmap cannot determine whether the port is open because packet filtering (e.g., a firewall, router rules etc.) prevents its probes from reaching the port.
4. **unfiltered:** A port is accessible (using ACK), but Nmap is unable to determine whether it is open or closed.
5. **open|filtered:** Nmap places ports in this state when it is unable to determine whether a port is open or filtered (This occurs for scan types in which open ports give no response.).
6. **closed|filtered:** This state is used when Nmap is unable to determine whether a port is closed or filtered.

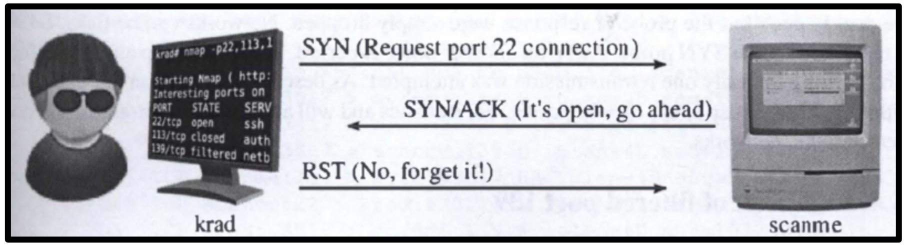
Nmap

Port Scanning technique:

❑ **TCP SYN Scan (-sS):** sends TCP packets to the targets on specific port.

- do not creates a full TCP session (instead send RST after the SYN/ACK message).
- requires root privileges.
- the default port scanning method.

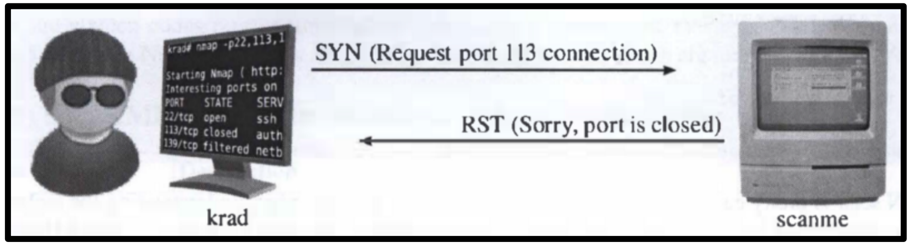
Example (1): TCP SYN scan on OPEN port



```
krad# nmap -p22,113,1
Starting Nmap ( http:
Interesting ports on
PORT    STATE    SERV
22/tcp  open     ssh
113/tcp closed   auth
139/tcp filtered netb
```

SYN (Request port 22 connection)
SYN/ACK (It's open, go ahead)
RST (No, forget it!)
krad
scanme

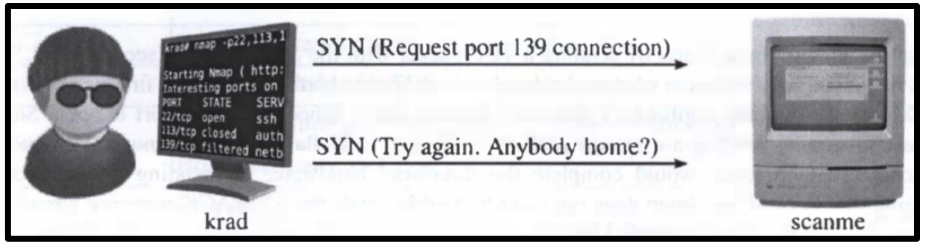
Example (2): TCP SYN scan on CLOSED port



```
krad# nmap -p22,113,1
Starting Nmap ( http:
Interesting ports on
PORT    STATE    SERV
22/tcp  open     ssh
113/tcp closed   auth
139/tcp filtered netb
```

SYN (Request port 113 connection)
RST (Sorry, port is closed)
krad
scanme

Example (3): TCP SYN scan on FILTERED port



```
krad# nmap -p22,113,1
Starting Nmap ( http:
Interesting ports on
PORT    STATE    SERV
22/tcp  open     ssh
113/tcp closed   auth
139/tcp filtered netb
```

SYN (Request port 139 connection)
SYN (Try again. Anybody home?)
krad
scanme

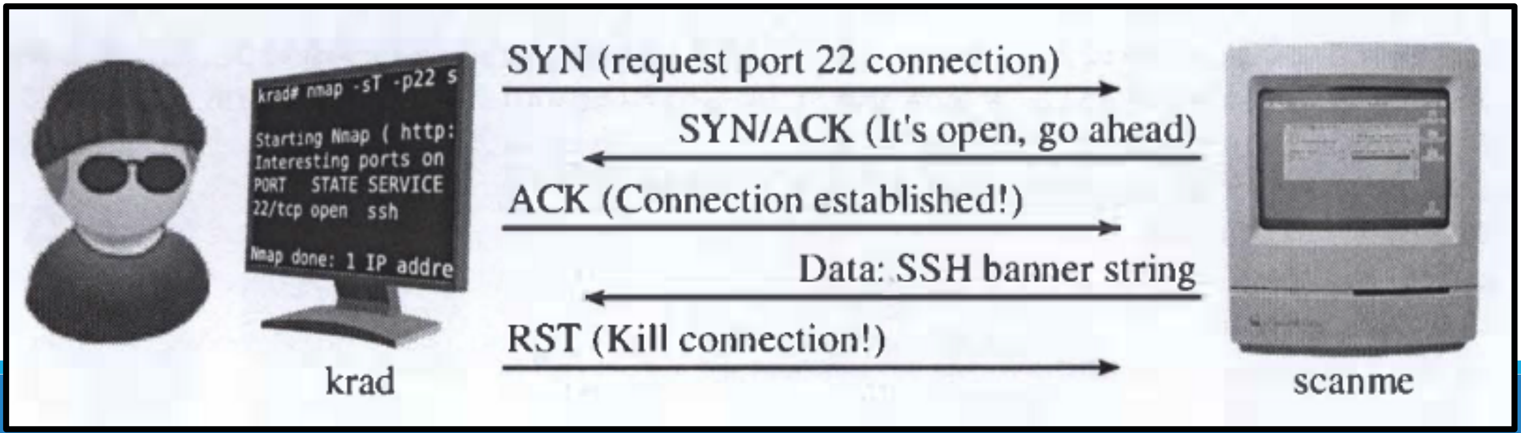
Nmap

Port Scanning technique:

❑ **TCP connect scan (-sT):** establish a connection with the target server by issuing the “connect” system call.

- targets are more likely to allow the connection because it tries to establish a connection with target same as network.
- scan takes more time to complete and requires to generate more packets.
- does not requires root privileges.

Example: TCP connect scan on OPEN port



```
krad# nmap -sT -p22 s

Starting Nmap ( http:
Interesting ports on
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP addre
```

SYN (request port 22 connection)
SYN/ACK (It's open, go ahead)
ACK (Connection established!)
Data: SSH banner string
RST (Kill connection!)
krad
scanme

Nmap

Port Scanning technique:

❑ **UDP Scan (-sU):** sends UDP packets in order to scan UDP ports (e.g, DNS, DHCP), can be combined with TCP SYN scan.

- generally slower and more difficult than TCP.
- sends a UDP packet to **every** targeted port.
- for most ports the packet is empty (however, for some common ports such as 53, a protocol-specific payload is sent to increase response rate).

Nmap

Port Scanning technique:

- **Null, Xmas and FIN Scan (-sN, -sX and -sF):** These three scan types exploit a loophole in the **TCP RFC** to differentiate between open and closed ports.

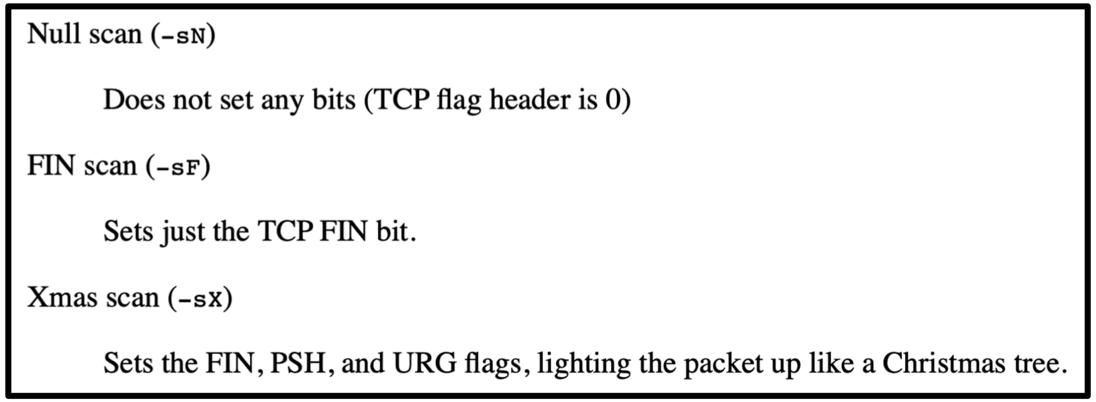
  (*) TCP RFC says that any packet not containing SYN, RST, or ACK bits will result in a returned RST if the port is closed and no response at all if the port is open.

  (**) These three scan types are exactly the same in behavior except for the TCP flags set in probe packets (see picture).

  (***) If a RST packet is received, the port is considered closed, while no response means it is open or filtered (The port is marked filtered if an ICMP unreachable error is received).

Pros. The key advantage to these scan types is that they can sneak through certain non-stateful firewalls and packet filtering routers (however, most modern IDS products can be configured to detect them).

Cons. The big downside is that not all systems follow the TCP RFC.



Null scan (`-sN`)

Does not set any bits (TCP flag header is 0)

FIN scan (`-sF`)

Sets just the TCP FIN bit.

Xmas scan (`-sX`)

Sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree.

Nmap

Service, version and OS Detection

Service and version detection (-sV/-A) :

- Nmap contains a database of fingerprints for about 2,200 well-known services.
- Nmap would report that those ports probably correspond to a mail server (SMTP), web server (HTTP), and name server (DNS) respectively - this lookup is usually accurate.
- In addition, Nmap can derive the version number of service, which helps dramatically in determining which exploits a server is vulnerable to.

OS detection (-O) :

- One of Nmap's best-known features is remote OS detection using TCP/IP stack fingerprinting.
- Nmap sends a series of TCP and UDP packets to the remote host and examines practically every bit in the responses.
- After performing dozens of tests (such as TCP options support, the initial window size etc.), Nmap compares the results to its with a database of more than 2,600 known OS fingerprints.
- Each fingerprint includes a freeform textual description of the OS.

Nmap

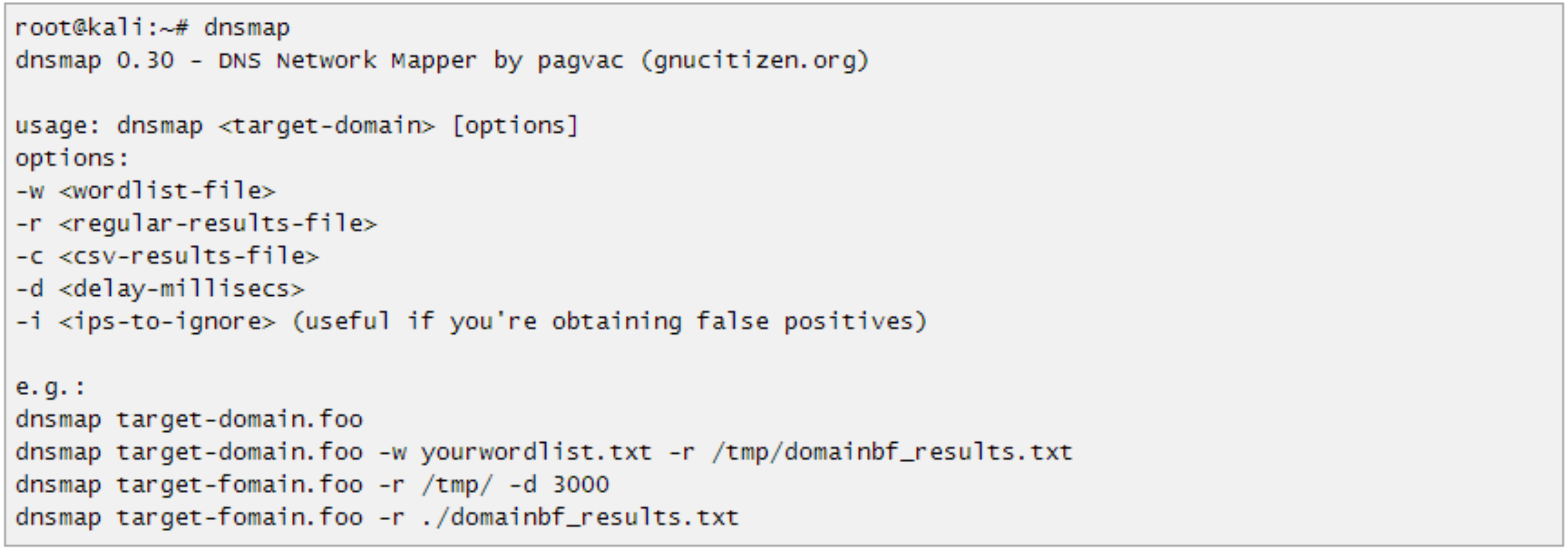
Additional interesting flags:

-p <port ranges> : specifics which ports you want to scan.

--exclude-ports *<port ranges>* : this option specifies which ports you do want Nmap to exclude from scanning.

dnsmap

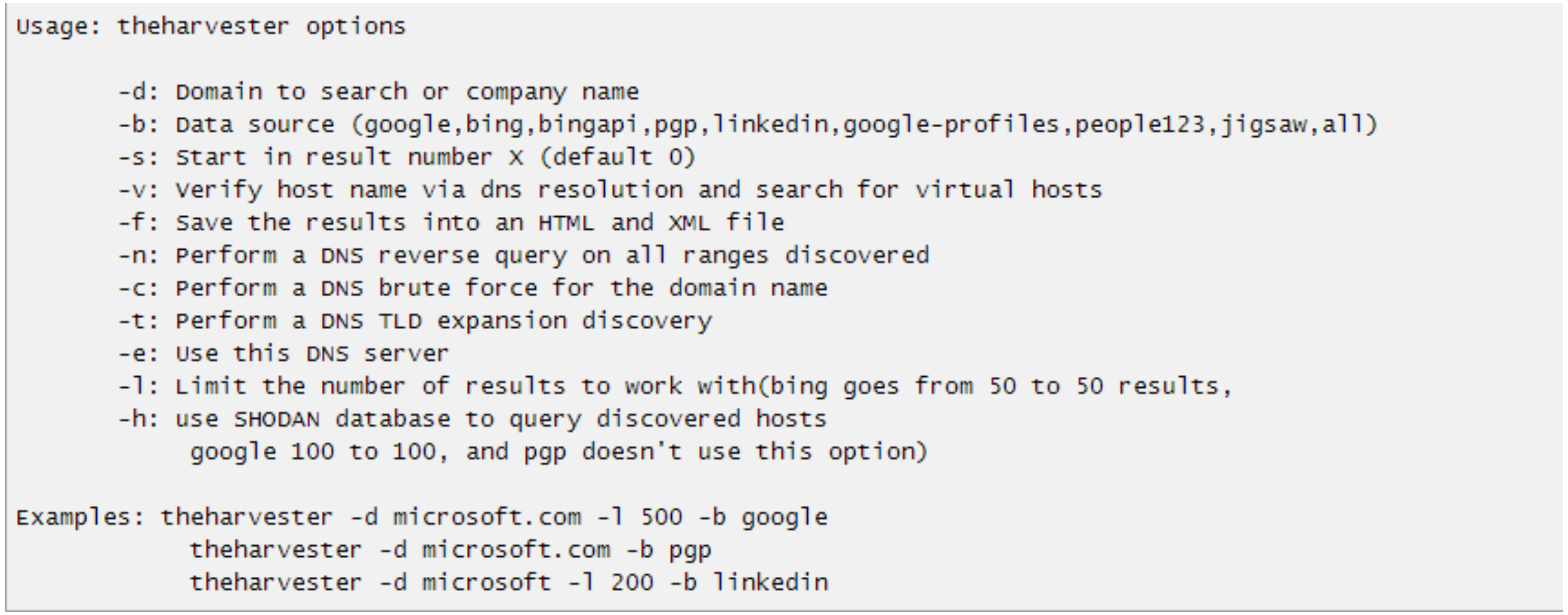
- Meant to be used by pentesters during the information gathering of infrastructure security asses.
- Finds IP addresses and sub-domains of a given domain using brute-force techniques.



```
root@kali:~# dnsmap
dnsmap 0.30 - DNS Network Mapper by pagvac (gnucitizen.org)

usage: dnsmap <target-domain> [options]
options:
-w <wordlist-file>
-r <regular-results-file>
-c <csv-results-file>
-d <delay-millisecs>
-i <ips-to-ignore> (useful if you're obtaining false positives)

e.g.:
dnsmap target-domain.foo
dnsmap target-domain.foo -w yourwordlist.txt -r /tmp/domainbf_results.txt
dnsmap target-fomain.foo -r /tmp/ -d 3000
dnsmap target-fomain.foo -r ./domainbf_results.txt
```

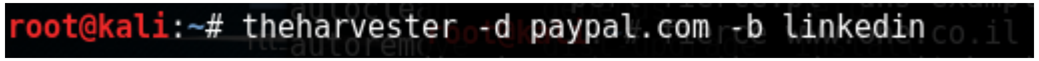
theharvester

- Help penetration testers in the early stages of the penetration test in order to understand the customer footprint on the internet
- Gather emails, subdomains, hosts, employee names, open port etc.
- Gather info from various sources : Google, LinkedIn, Twitter etc.

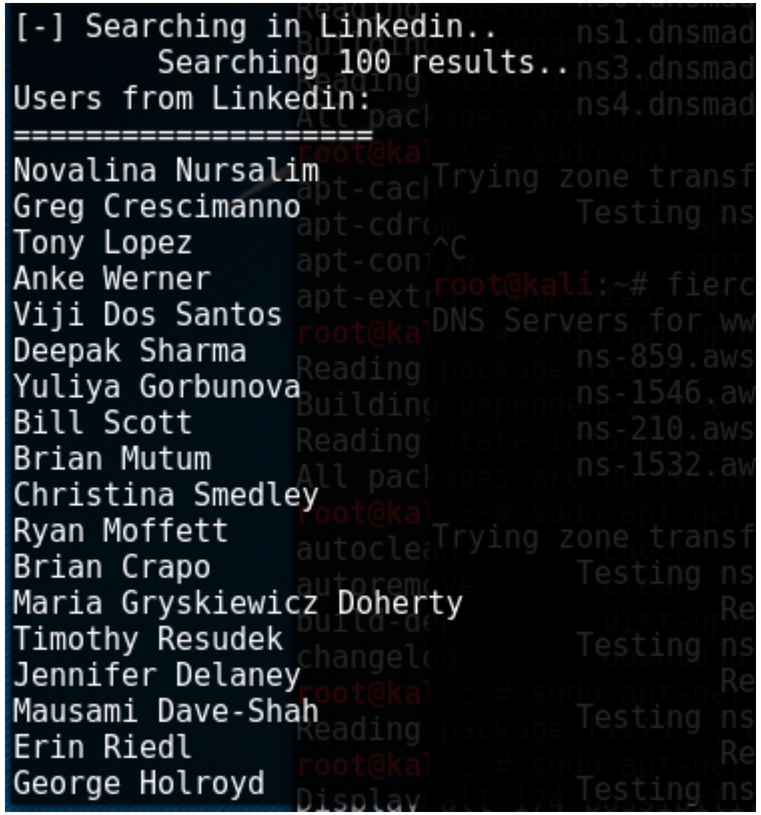


```
Usage: theharvester options

       -d: Domain to search or company name
       -b: Data source (google,bing,bingapi,pgp,linkedin,google-profiles,people123,jigsaw,all)
       -s: Start in result number X (default 0)
       -v: Verify host name via dns resolution and search for virtual hosts
       -f: Save the results into an HTML and XML file
       -n: Perform a DNS reverse query on all ranges discovered
       -c: Perform a DNS brute force for the domain name
       -t: Perform a DNS TLD expansion discovery
       -e: Use this DNS server
       -l: Limit the number of results to work with(bing goes from 50 to 50 results,
       -h: use SHODAN database to query discovered hosts
            google 100 to 100, and pgp doesn't use this option)

Examples: theharvester -d microsoft.com -l 500 -b google
            theharvester -d microsoft.com -b pgp
            theharvester -d microsoft -l 200 -b linkedin
```

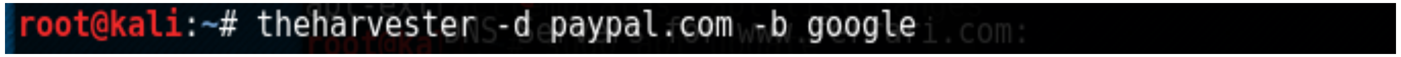
Theharvester Usage Example



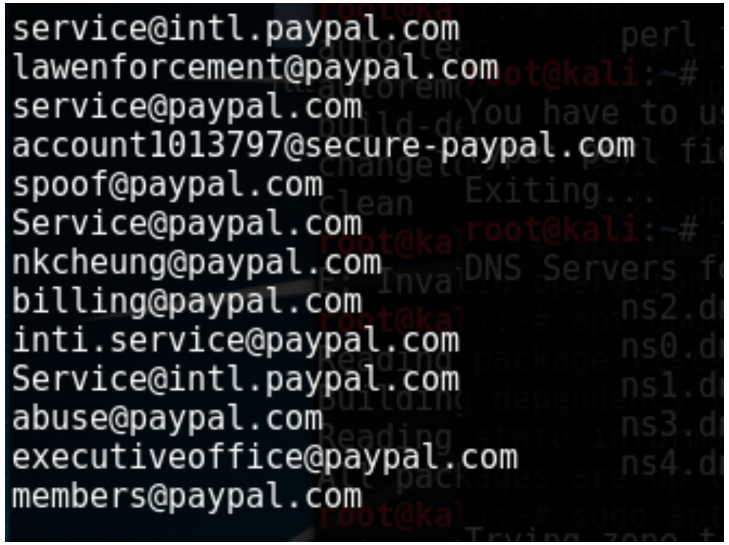
```
root@kali:~# theharvester -d paypal.com -b linkedin
```



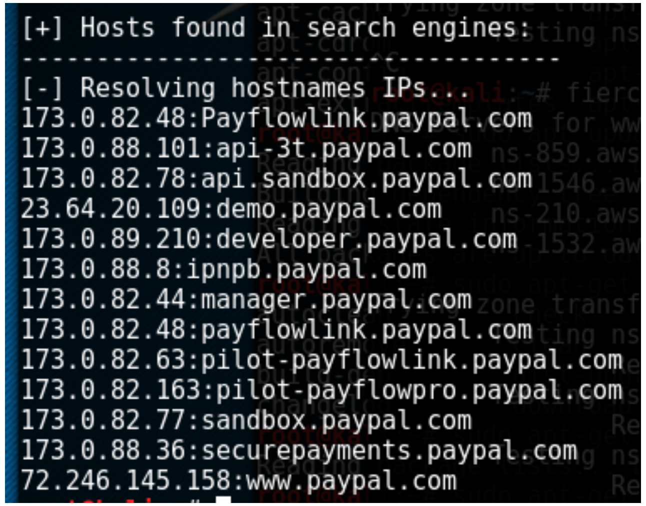
```
[-] Searching in Linkedin..
        Searching 100 results..
Users from Linkedin:
====================
Novalina Nursalim
Greg Crescimanno
Tony Lopez
Anke Werner
Viji Dos Santos
Deepak Sharma
Yuliya Gorbunova
Bill Scott
Brian Mutum
Christina Smedley
Ryan Moffett
Brian Crapo
Maria Gryskiewicz Doherty
Timothy Resudek
Jennifer Delaney
Mausami Dave-Shah
Erin Riedl
George Holroyd
```



```
root@kali:~# theharvester -d paypal.com -b google
```



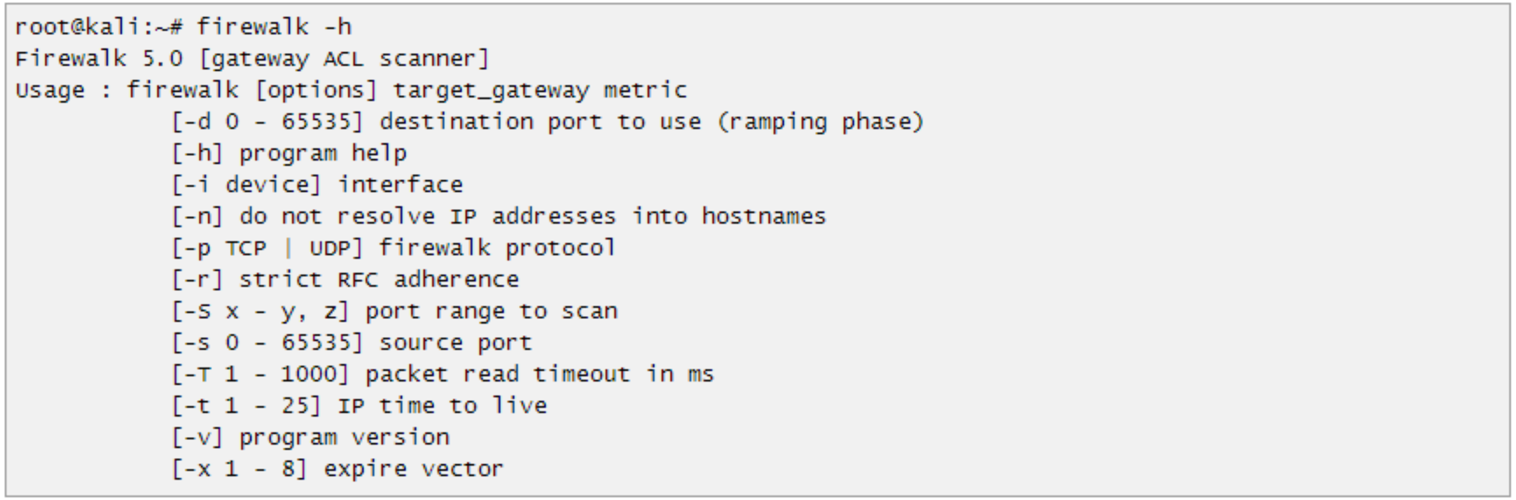
```
service@intl.paypal.com
lawenforcement@paypal.com
service@paypal.com
account1013797@secure-paypal.com
spoof@paypal.com
Service@paypal.com
nkcheung@paypal.com
billing@paypal.com
inti.service@paypal.com
Service@intl.paypal.com
abuse@paypal.com
executiveoffice@paypal.com
members@paypal.com
```



```
[+] Hosts found in search engines:
------------------------------------
[-] Resolving hostnames IPs...
173.0.82.48:Payflowlink.paypal.com
173.0.88.101:api-3t.paypal.com
173.0.82.78:api.sandbox.paypal.com
23.64.20.109:demo.paypal.com
173.0.89.210:developer.paypal.com
173.0.88.8:ipnpb.paypal.com
173.0.82.44:manager.paypal.com
173.0.82.48:payflowlink.paypal.com
173.0.82.63:pilot-payflowlink.paypal.com
173.0.82.163:pilot-payflowpro.paypal.com
173.0.82.77:sandbox.paypal.com
173.0.88.36:securepayments.paypal.com
72.246.145.158:www.paypal.com
```

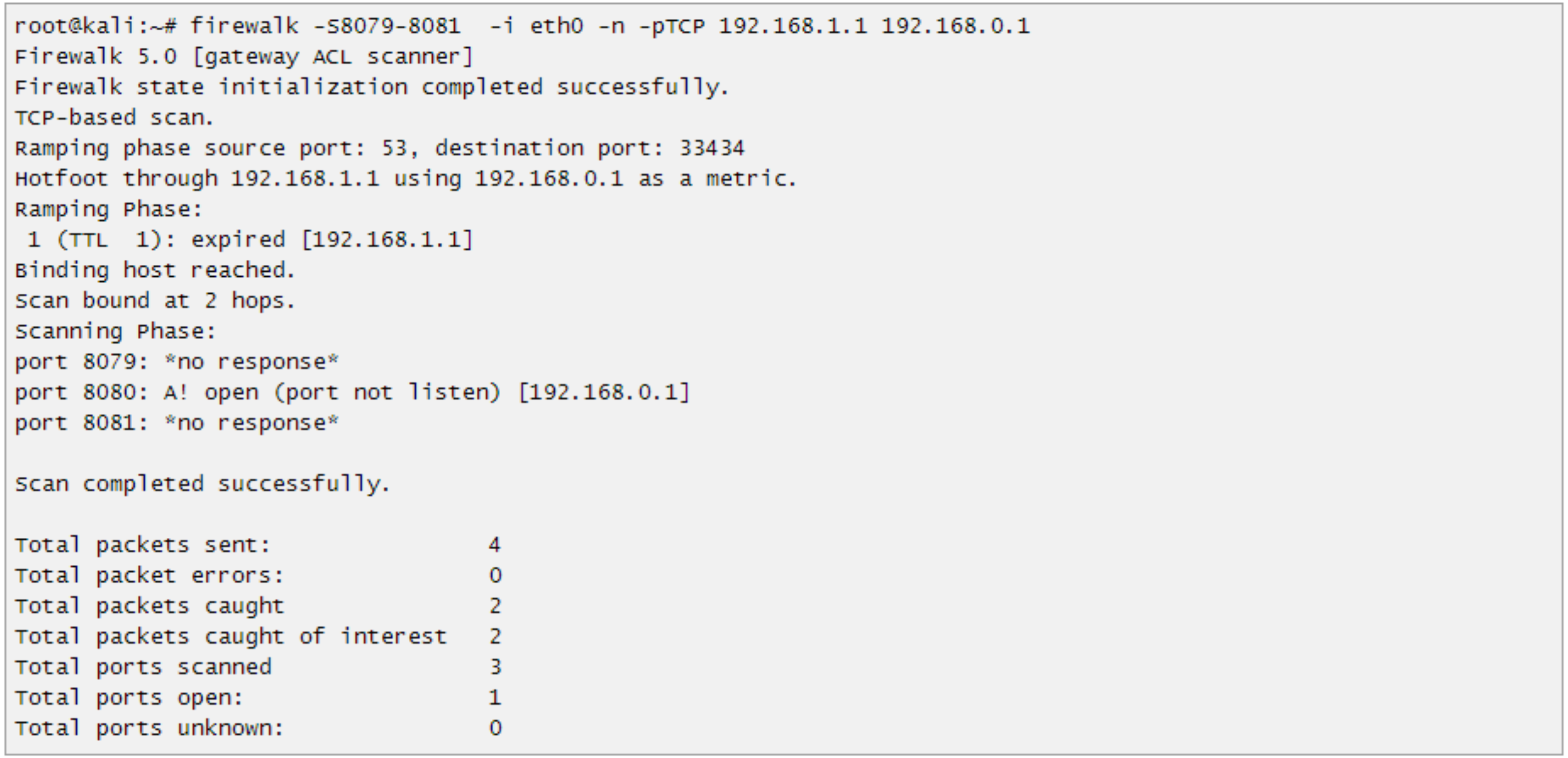
Firewalk

- ❑ An active reconnaissance network security tool that attempts to determine what layer 4 protocols a given IP forwarding device will pass (TCP/UDP)
- ❑ Firewalk works by sending out TCP or UDP packets with a TTL one greater than the targeted gateway
  - If the gateway allows the traffic, it will forward the packets to the next hop where they will expire and elicit an ICMP_TIME_EXCEEDED message. Else, it will likely drop the packets.



```
root@kali:~# firewalk -h
Firewalk 5.0 [gateway ACL scanner]
Usage : firewalk [options] target_gateway metric
           [-d 0 - 65535] destination port to use (ramping phase)
           [-h] program help
           [-i device] interface
           [-n] do not resolve IP addresses into hostnames
           [-p TCP | UDP] firewalk protocol
           [-r] strict RFC adherence
           [-S x - y, z] port range to scan
           [-s 0 - 65535] source port
           [-T 1 - 1000] packet read timeout in ms
           [-t 1 - 25] IP time to live
           [-v] program version
           [-x 1 - 8] expire vector
```

Firewalk – Usage Example

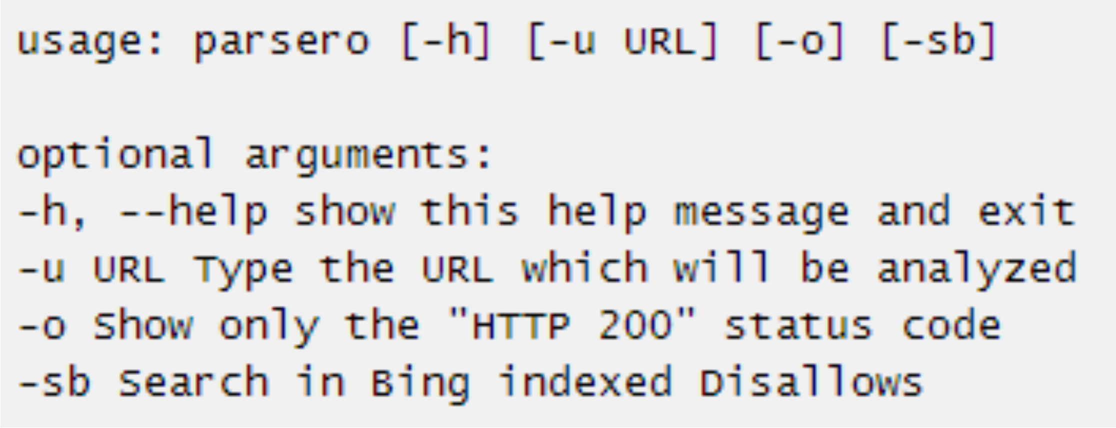


```
root@kali:~# firewalk -S8079-8081  -i eth0 -n -pTCP 192.168.1.1 192.168.0.1
Firewalk 5.0 [gateway ACL scanner]
Firewalk state initialization completed successfully.
TCP-based scan.
Ramping phase source port: 53, destination port: 33434
Hotfoot through 192.168.1.1 using 192.168.0.1 as a metric.
Ramping Phase:
 1 (TTL  1): expired [192.168.1.1]
Binding host reached.
Scan bound at 2 hops.
Scanning Phase:
port 8079: *no response*
port 8080: A! open (port not listen) [192.168.0.1]
port 8081: *no response*

Scan completed successfully.

Total packets sent:                4
Total packet errors:               0
Total packets caught               2
Total packets caught of interest   2
Total ports scanned                3
Total ports open:                  1
Total ports unknown:               0
```

Parsero

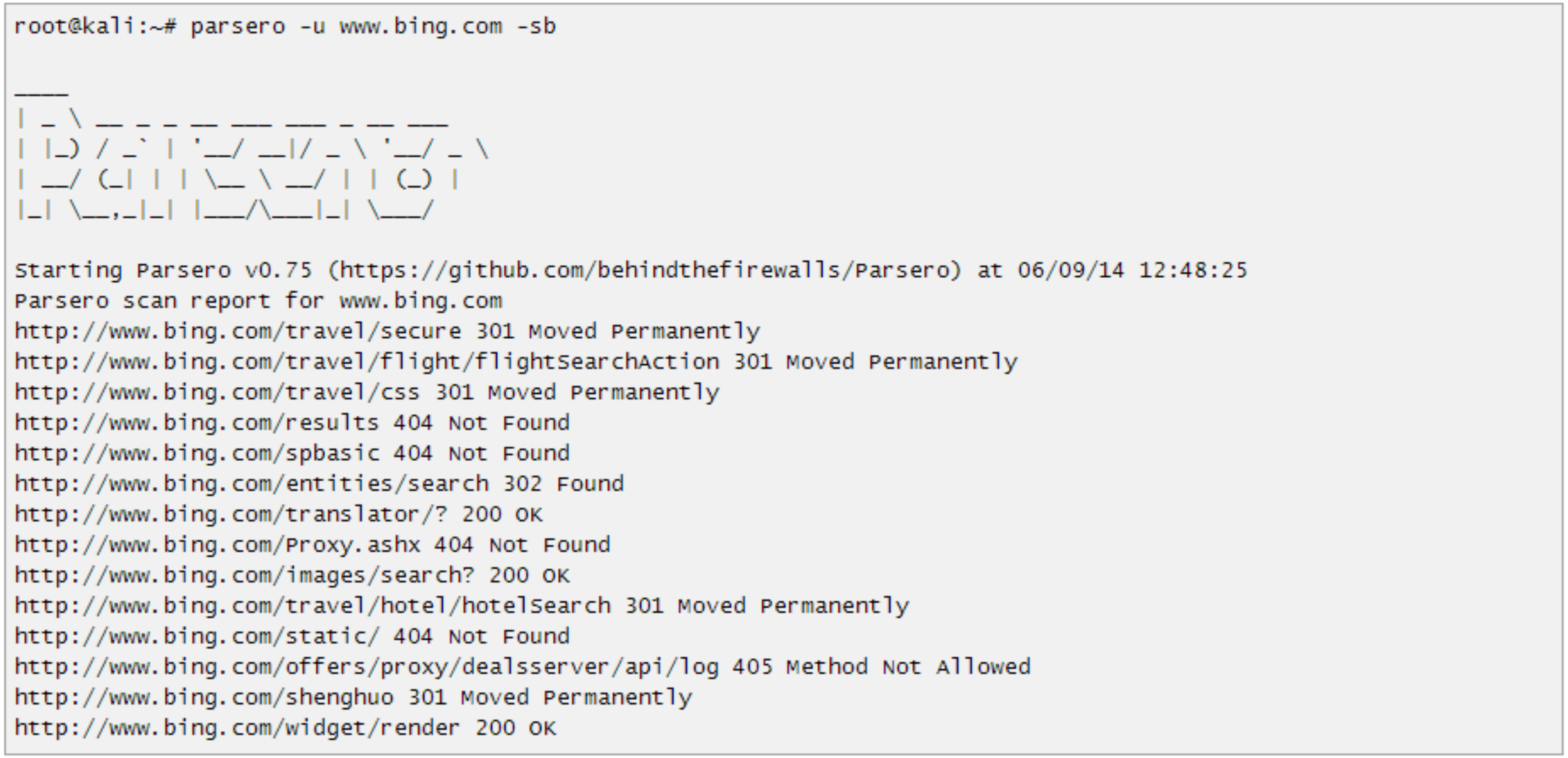
- Free script written in Python
- Parsero reads Robot.txt file of a web server and looks at the Disallow entries

  For Example: “Disallow /portal/login_access”
- Disallow entries means that the content of the entries is not allowed to be indexed by crawlers like Google, Bing, Yahoo etc.



```
usage: parsero [-h] [-u URL] [-o] [-sb]

optional arguments:
-h, --help show this help message and exit
-u URL Type the URL which will be analyzed
-o Show only the "HTTP 200" status code
-sb Search in Bing indexed Disallows
```

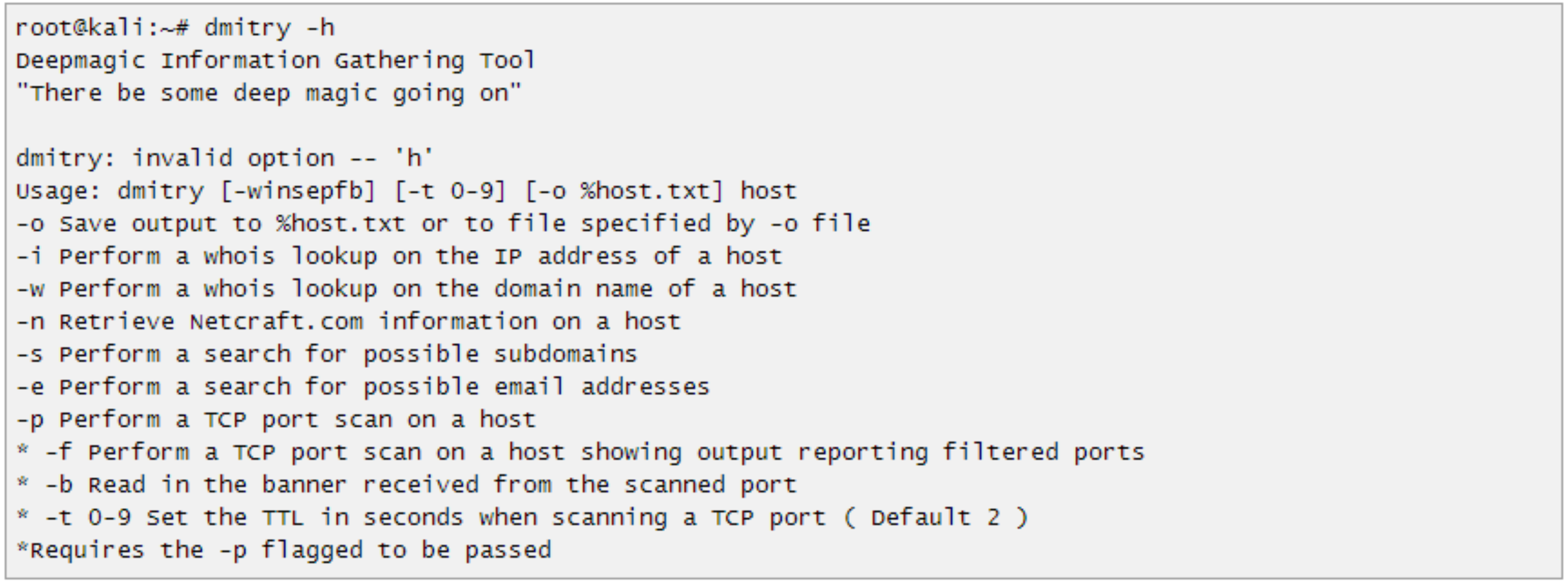
Parsero – Usage Example



```
root@kali:~# parsero -u www.bing.com -sb


____
| _ \ __ _ _ __ ___ ___ _ __ ___
| |_) / _` | '__/ __|/ _ \ '__/ _ \
| __/ (_| | | \__ \ __/ | | (_) |
|_| \__,_|_| |___/\___|_| \___/

Starting Parsero v0.75 (https://github.com/behindthefirewalls/Parsero) at 06/09/14 12:48:25
Parsero scan report for www.bing.com
http://www.bing.com/travel/secure 301 Moved Permanently
http://www.bing.com/travel/flight/flightSearchAction 301 Moved Permanently
http://www.bing.com/travel/css 301 Moved Permanently
http://www.bing.com/results 404 Not Found
http://www.bing.com/spbasic 404 Not Found
http://www.bing.com/entities/search 302 Found
http://www.bing.com/translator/? 200 OK
http://www.bing.com/Proxy.ashx 404 Not Found
http://www.bing.com/images/search? 200 OK
http://www.bing.com/travel/hotel/hotelSearch 301 Moved Permanently
http://www.bing.com/static/ 404 Not Found
http://www.bing.com/offers/proxy/dealsserver/api/log 405 Method Not Allowed
http://www.bing.com/shenghuo 301 Moved Permanently
http://www.bing.com/widget/render 200 OK
```

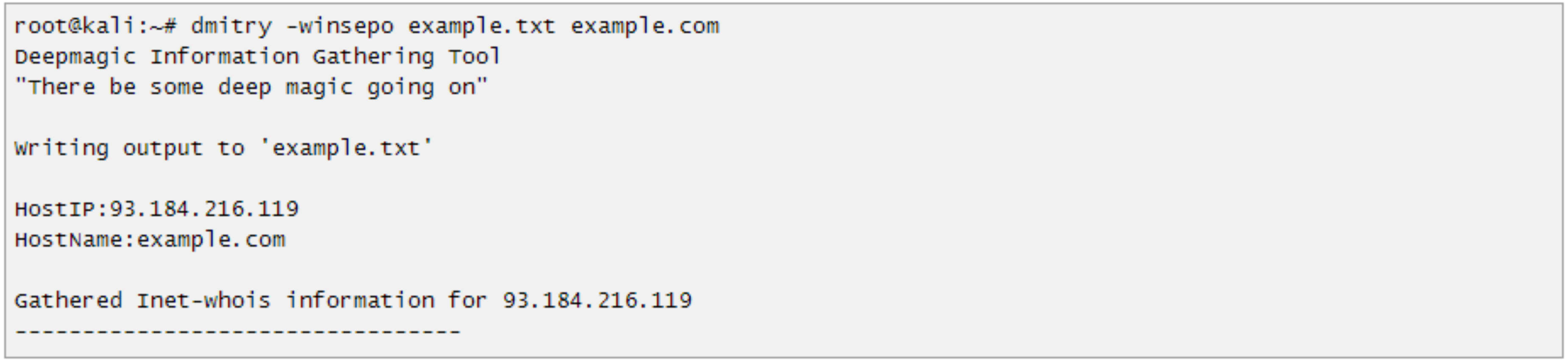
DMitry

- ❑ Linux command line application
- ❑ Has the ability to gather as much information as possible about a host
  - Sub-domains
  - Email addresses
  - Uptime Info.
  - TCP port scan
  - Whois lookups



```
root@kali:~# dmitry -h
Deepmagic Information Gathering Tool
"There be some deep magic going on"

dmitry: invalid option -- 'h'
Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
-o Save output to %host.txt or to file specified by -o file
-i Perform a whois lookup on the IP address of a host
-w Perform a whois lookup on the domain name of a host
-n Retrieve Netcraft.com information on a host
-s Perform a search for possible subdomains
-e Perform a search for possible email addresses
-p Perform a TCP port scan on a host
* -f Perform a TCP port scan on a host showing output reporting filtered ports
* -b Read in the banner received from the scanned port
* -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
```

Dmitry – Usage Example



```
root@kali:~# dmitry -winsepo example.txt example.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Writing output to 'example.txt'

HostIP:93.184.216.119
HostName:example.com

Gathered Inet-whois information for 93.184.216.119
---------------------------------
```

Other Information gathering Tools

- ✓ Previous tools documentation
- ✓ Other powerful Info. Gathering tools

http://tools.kali.org/category/information-gathering

Open Kali Linux (LAB)

In order to start Kali Linux in the labs follow the next steps:

1. Open ‘VMware Player’

vmplayer.exe
VMware Player
VMware, Inc.

2. Choose ‘Player’

3. Choose ‘File’

4. Choose ‘Open a virtual machine’

5. Go to ‘E:/VM-s/Kali/’ and choose the VM file

<u>Kali Linux</u>

- Username: root
- Password : toor

Lab Exercise – one week to submission

1. Make list of 5 possible sub-domains for **ise.bgu.ac.il** and scan the lists to see weather the sub-domain exists. Please append to the answer file both the list and the scan output.

2. What IP addresses can be pinged in the subnet 132.72.81.1/26 Notice IP address 132.72.81.35

- Is the host alive?
- Which TCP and UDP ports are opened?
- What is the OS version on that device?
- What services are available (use version scan)?

3. Gather 5 emails addresses and 5 sub-domains of **bgu.ac.il**