Inguma module library documentation

- Discover modules
  - arping
  - hostname
  - ipaddr
  - ping
  - isnated
  - ispromisc
  - trace
  - whois
- Gather modules
  - example
  - apps11i
  - nids
  - nmapfp
  - oratool
  - osifuzz
  - p0f
  - portscan
  - rpcdump
  - samrdump
  - identify
  - smbclient
  - sniffer
  - snmpwalk
  - winspdetect
  - tcpscan
  - tnscmd
  - winspdetect
  - xmlrpc
  - rpceminfo
  - oascheck
- Brute modules
  - bruteftp
  - bruteimap
  - brutepop
  - brutesmb
  - brutesyb
- Fuzzing modules
  - ftpfuzz

Discover modules

arping

The module sends an arp who has message to discover hosts. IP addresses as well as ranges can be used.

It requires root or administrator privileges depending on the operative system because the module will use raw sockets.

Arguments

target: The target network or host.

timeout: Timeout.

Example

```
inguma> target = "192.168.1.0/24"
inguma> arping
Adding to discovered hosts 192.168.1.1
Adding to discovered hosts 192.168.1.12
Adding to discovered hosts 192.168.1.14
Adding to discovered hosts 192.168.1.21

List of discovered hosts
------------------------

00:13:49:e1:9b:c0 192.168.1.1 (ZyXEL Communications)
00:0a:e6:18:07:45 192.168.1.12 (Elitegroup Computer System Co. (ECS))
00:0c:29:7f:54:5a 192.168.1.14 (VMware)
00:0c:29:fa:87:7f 192.168.1.21 (Vmware)
```

hostname

Returns the associated hostname for the **target**.

Arguments

target: The target host alias or ip address.

Example

```
inguma> target = “192.168.1.11”
inguma> hostname
joxeanbox01
```

ipaddr

Returns the ip address associated to a hostname or host alias.

Arguments

target: The target hostname, host alias or ip address.

ping

Try pinging host or network. Hostnames, ip addresses as well as ranges can be used.

Arguments

target: The target network or host.

timeout: Timeout of the operation.

packetType: Type of ICMP packet to send. By default ECHO_REQUEST. The following list enumerates the possible packet types:

- ECHO_REPLY = 0
- DEST_UNREACH = 3
- SOURCE_QUENCH = 4
- REDIRECT = 5
- ECHO_REQUEST = 8
- ROUTER_ADVERTISEMENT = 9
- ROUTER_SOLICITATION = 10
- TIME_EXCEEDED = 11
- PARAMETER_PROBLEM = 12
- TIMESTAMP_REQUEST = 13
- TIMESTAMP_REPLY = 14
- INFORMATION_REQUEST = 15
- INFORMATION_RESPONSE = 16
- ADDRESS_MASK_REQUEST = 17
- ADDRESS_MASK_REPPLY = 18

Example

```
inguma> target = "192.168.1.10-20"
inguma> timeout = 0.1
inguma> ping
WARNING: Mac address to reach 192.168.1.10 not found

Adding to alive hosts 192.168.1.12
WARNING: Mac address to reach 192.168.1.13 not found

Adding to alive hosts 192.168.1.14
```

```
WARNING: Mac address to reach 192.168.1.15 not found

WARNING: Mac address to reach 192.168.1.16 not found

WARNING: Mac address to reach 192.168.1.17 not found

WARNING: Mac address to reach 192.168.1.18 not found

WARNING: Mac address to reach 192.168.1.19 not found

WARNING: Mac address to reach 192.168.1.20 not found


Discovered hosts
----------------

Found host 1 192.168.1.12
Found host 2 192.168.1.14
```

isnated

Checks if one opened port at host is NATed or not by sending first an ICMP packet to get the number of hops and, next, compare it with the number of hops by sending a TCP packet.

Arguments

target: The target host.

timeout: Timeout.

Example

```
inguma> target = "www.microsoft.com"; port = 80
inguma> isnated
Port 80 is NOT NATed
inguma> target = “xxx.xxx.com”
inguma> isnated
Port 80 is NATed
```

ispromisc

Checks if the target is in promiscuous state.

Arguments

target: The target host or ip address.

Example

```
inguma> target = "192.168.1.21"
inguma> ispromisc
False
inguma> target = "192.168.1.1"
inguma> ispromisc
True
```

trace

Trace the route to the host. Similar to the traceroute tool.

The tool requires root or administrator privileges.

Arguments

minttl: Minimun TTL.

maxttl: Maximun TTL.

sport: Source port.

dport: Destination port.

timeout: Timeout.

Example

```
inguma> target = "www.google.com"
inguma> trace
Adding hop  192.168.1.11
Adding hop  10.0.58.158
Adding hop  149.6.132.1
Adding hop  130.117.1.137
Adding hop  209.85.252.40
Adding hop  72.14.233.63
Adding hop  130.117.1.78
Adding hop  64.233.183.103
Adding hop  216.239.43.30

Trace to target(s)
------------------

Hop 1 192.168.1.11
Hop 2 10.0.58.158
Hop 3 149.6.132.1
Hop 4 130.117.1.137
Hop 5 209.85.252.40
Hop 6 72.14.233.63
Hop 7 130.117.1.78
```

```
Hop 8 64.233.183.103
Hop 9 216.239.43.30
Hop 10 64.233.183.103
```

whois

Check the whois database.

Arguments

target: Target hostname.