masscanned

_Frky

Sep 25, 2023

USAGE:

CHAPTER

ONE

INTRODUCTION

Masscanned is a low-interaction honeypot, primarily designed to help gather intelligence about network scanners and bots.

It has been built as a companion tool for IVRE but can be used independently.

The code is on GitHub.

Here is a quick demo:

CHAPTER

TWO

STATUS OF THIS DOCUMENTATION

This documentation is a work in progress!

CHAPTER

THREE

CONTENT

3.1 Using Masscanned

3.1.1 Dedicated addresses

Masscanned is designed to handle its own IP addresses, which means that the host should not have those addresses configured, and Masscanned will answer `ARP` requests (or `ICMPv6 ND` neighbor sollicitations).

The host may have one or more (`IPv4` and/or `IPv6`) addresses configured on an interface also used by masscanned, but those addresses must be different from those configured to be used by masscanned.

In that situation (dedicated addresses), just run:

```
# masscanned -i <iface> -f <ip_addr_file>
```

where <`ip_addr_file`> is the path of a text file with one address (`IPv4` or `IPv6`) per line.

3.1.2 Addresses shared with the host

Sometimes it is desirable to have an IP address used by the host (*e.g.*, for administration tasks) and by masscanned (to handle all other incoming packets).

Since this is not implemented in masscanned, a tiny hack is needed: we are going to run it on a `veth` interface.

For this example, we suppose:

- The interface is `eth0`, the address is `192.168.0.10`.
- We want masscanned to handle all the traffic except for incoming SSH connections on TCP/22 port.

We create a `veth` pair of interfaces, on which we are going to use the 0.255.0.0/31 network (which should not be a problem since 0.0.0.0/8 is reserved as "Current Network"):

```
# ip link add to_masscanned type veth peer masscanned
# ip link set masscanned up
# ip link set to_masscanned up
# ip addr add 0.255.0.0/31 dev to_masscanned
# masscanned -i masscanned
```

Masscanned can now be used, but only from the host where it runs:

```
# ping -c 1 0.255.0.1
PING 0.255.0.1 (0.255.0.1) 56(84) octets de données.
64 octets de 0.255.0.1 : icmp_seq=1 ttl=64 temps=0.442 ms

--- statistiques ping 0.255.0.1 ---
1 paquets transmis, 1 reçus, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.442/0.442/0.442/0.000 ms
```

Now, we are going to use Netfilter / `iptables` to redirect incoming traffic to masscanned:

```
# sysctl -w net.ipv4.ip_forward=1
# iptables -t nat -A PREROUTING -i eth0 -d 192.168.0.10 -p tcp --dport 22 -j ACCEPT
# iptables -t nat -A PREROUTING -i eth0 -d 192.168.0.10/32 -j DNAT --to-destination 0.
↪255.0.1
```

And, from another host on the 192.168.0.0/24 network:

```
# ping -c 1 192.168.0.10
PING 192.168.0.10 (192.168.0.10) 56(84) octets de données.
64 octets de 192.168.0.10 : icmp_seq=1 ttl=63 temps=0.366 ms

--- statistiques ping 192.168.0.10 ---
1 paquets transmis, 1 reçus, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.366/0.366/0.366/0.000 ms
```

The masscanned output:

```
WARN - ARP-Reply to ea:c0:d6:20:0c:6a for IP 0.255.0.1
WARN - ICMP-Echo-Reply to ICMP-Echo-Request
```