Morpheus

Information Security Inc.

iSEC
information security inc.

Contents

- About Morpheus
- Dependencies
- Demo Setup
- Installing Morpheus
- Updating Morpheus
- Using Morpheus
- References

About Morpheus

- Morpheus it's a Man-In-The-Middle (mitm) suite that allows users to manipulate tcp/udp data using ettercap, urlsnarf, msgsnarf and tcpkill as backend applications

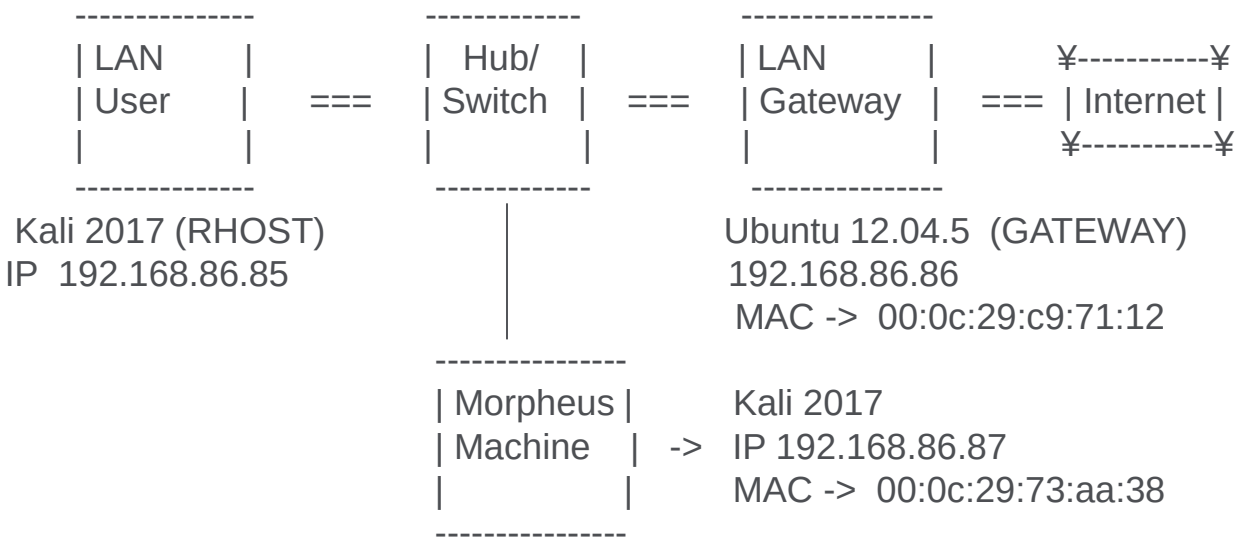
Dependencies

- required: ettercap, nmap, zenity, apache2
- sub-dependencies: driftnet, dsniff (urlsnarf,tcpkill,msgsnarf)

- All dependencies and sub-dependencies are met in Kali Linux 2017

Demo Setup

- Setup



Installing Morpheus

- Clone GitHub repo

```
root@kali2017:~# git clone https://github.com/r00t-3xp10it/morpheus
Cloning into 'morpheus'...
remote: Counting objects: 220, done.
remote: Total 220 (delta 0), reused 0 (delta 0), pack-reused 220
Receiving objects: 100% (220/220), 350.07 KiB | 791.00 KiB/s, done.
Resolving deltas: 100% (131/131), done.
root@kali2017:~# cd morpheus/
root@kali2017:~/morpheus# ls
bin  filters  logs  morpheus.sh  output  README.md  settings
```

Updating Morpheus

- Updating Morpheus

```
root@kali2017:~# cd morpheus/
root@kali2017:~/morpheus# git pull
Already up-to-date.
```

Using Morpheus

- Starting Morpheus

Using Morpheus

- Morpheus options

OPTION	DESCRIPTION(filters)
1	- Firewall filter (tcp/udp) - report/capture_creds
2	- Capture cookies (http/auth) - sidejacking attack
3	- Drop all packets (src/dst) - packets drop/kill
4	- Redirect browser traffic - to another domain
5	- Redirect browser traffic - to google sphere
6	- Sniff browser traffic (http) - visited url's
7	- Sniff browser traffic (http) - capture pictures
8	- Sniff chat messages (live) - AOL,IRC,YAHOO,MSN
9	- Inject backdoor into (</body>) - exe,bat,jar,ps1,dll
10	- Firefox browser heap-spray - buffer overflow
11	- Android browser heap-spray - buffer overflow
12	- Tor-browser heap-spray(windows) - buffer overflow
13	- Clone website + keylogger - javascript_keylogger
14	- Modem/router login webpage - javascript_keylooger
15	- Replace website images - img src=http://other
16	- Replace website text - replace: worlds
W	- Write your own filter
S	- Scan LAN for live hosts
H	- Morpheus github help
E	- Exit/close Morpheus

Using Morpheus

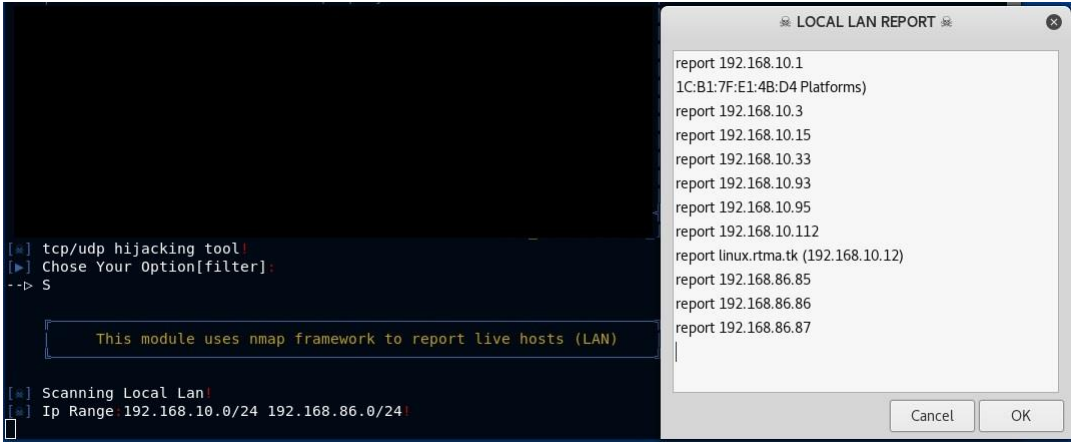
- Morpheus options

Using Morpheus

- Scanning LAN for live hosts

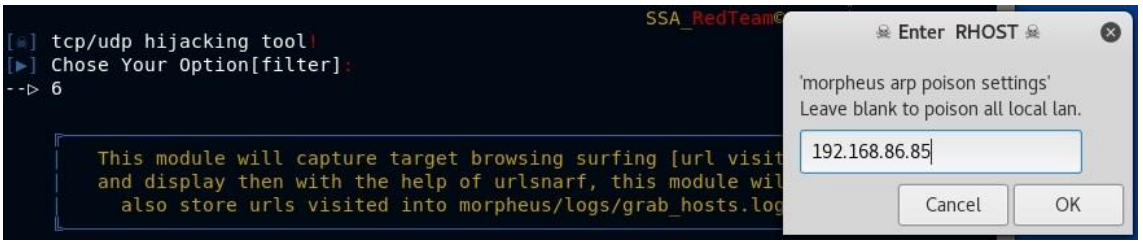
Using Morpheus

- Scanning LAN for live hosts



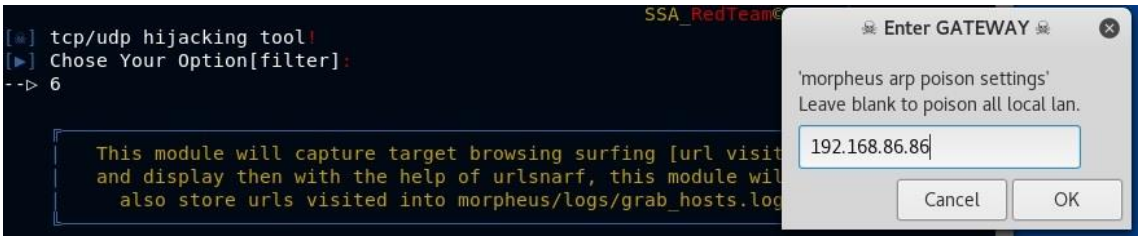
Using Morpheus

- Sniffing browser traffic (http)
- Configure the RHOST (the host to be ARP poisoned)



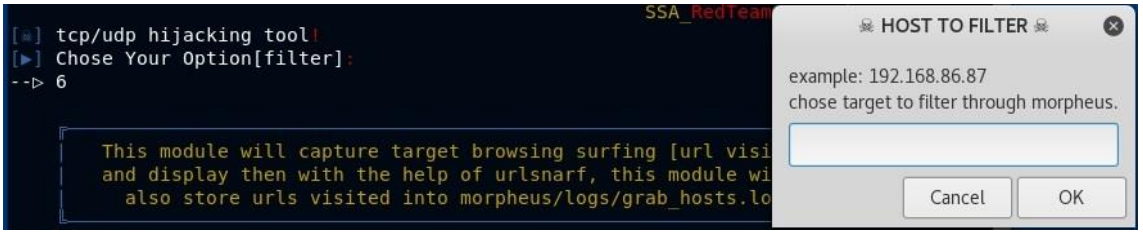
Using Morpheus

- Sniffing browser traffic (http)
- Configure the GATEWAY



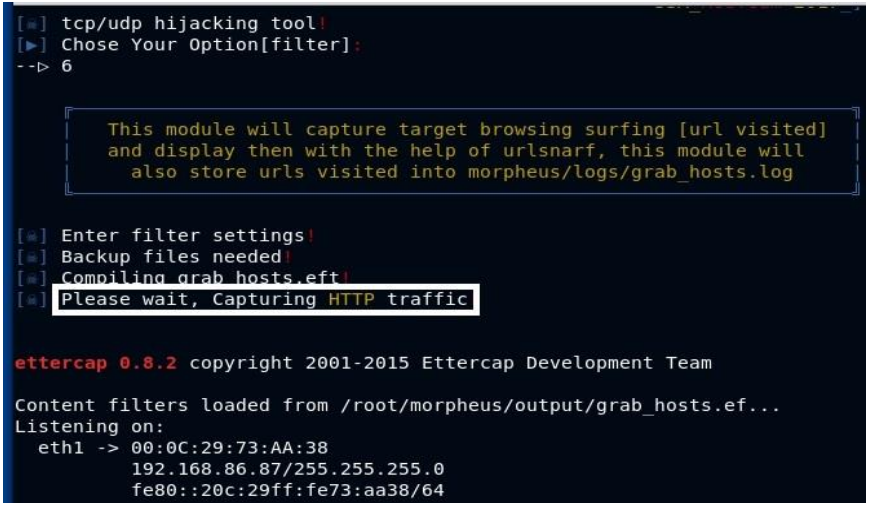
Using Morpheus

- Sniffing browser traffic (http)
- Configure the “HOST TO FILTER” (here its blank)



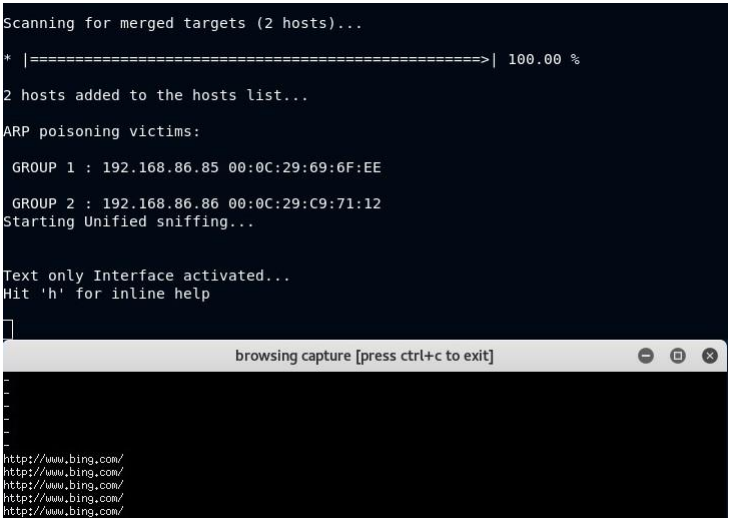
Using Morpheus

- Sniffing browser traffic (http)
- Capturing RHOST browser traffic



Using Morpheus

- Sniffing browser traffic (http)
- Capturing RHOST browser traffic (target website -> www.bing.com)



Using Morpheus

- Sniffing browser traffic (http)
- RHOST's arp and routing table **before** ARP poisoning

```
root@LUCKY64:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.86.87            ether   00:0c:29:73:aa:38   C                     eth4
192.168.86.86            ether   00:0c:29:c9:71:12   C                     eth4
192.168.10.1                     (incomplete)                              eth0
192.168.10.12                    (incomplete)                              eth0
192.168.10.15            ether   48:51:b7:15:98:cf   C                     eth0
root@LUCKY64:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.86.86   0.0.0.0         UG    0      0        0 eth4
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
192.168.10.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.86.0    0.0.0.0         255.255.255.0   U     0      0        0 eth4
```

Using Morpheus

- Sniffing browser traffic (http)
- RHOST's arp and routing table **after** ARP poisoning

```
root@LUCKY64:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.86.87            ether   00:0c:29:73:aa:38   C                     eth4
192.168.86.86            ether   00:0c:29:73:aa:38   C                     eth4
192.168.10.1             ether   1c:b1:7f:e1:4b:d4   C                     eth0
192.168.10.12                    (incomplete)                              eth0
192.168.10.15            ether   48:51:b7:15:98:cf   C                     eth0
root@LUCKY64:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.86.86   0.0.0.0         UG    0      0        0 eth4
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
192.168.10.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.86.0    0.0.0.0         255.255.255.0   U     0      0        0 eth4
```

Using Morpheus

- Sniffing browser traffic (http)
- Morpheus poisoning RHOST's ARP table

```
22:45:21.240520 ARP, Reply 192.168.86.86 is-at 00:0c:29:73:aa:38, length 46
        0x0000:  000c 2969 6fee 000c 2973 aa38 0806 0001  ..)io...)s.8....
        0x0010:  0800 0604 0002 000c 2973 aa38 c0a8 5656  ........)s.8..VV
        0x0020:  000c 2969 6fee c0a8 5655 0000 0000 0000  ..)io...VU......
        0x0030:  0000 0000 0000 0000 0000 0000            ............
```

References

- Kitploit
http://www.kitploit.com/2016/12/morpheus-automated-ettercap-tcpip.html

- Kali Linux
https://www.kali.org/downloads/

- Man-in-the-middle attack
https://en.wikipedia.org/wiki/Man-in-the-middle_attack

- ARP poisoning (spoofing)
https://en.wikipedia.org/wiki/ARP_spoofing

- Ettercap
https://github.com/Ettercap/ettercap