Names: PJ Sangvong, Luisa Escosteguy

Passive information gathering

- What domain did you investigate?
    - google.com
- What is its IP address?
    - 172.217.4.78
- When does the domain's registration expire?
    - 09/13/2028 (field Registry Expiry Date from whois google.com)
- What information, if any, did you learn about the people or corporation responsible for the domain in question? (Your answer could be less interesting than you had hoped due to the increasingly common use of domain privacy services. In that case, at least give me information about what you learned about the relevant domain privacy service.)
    - We got a lot of information, like some email addresses, phone, and dates but mostly nothing interesting. One mildly interesting fact is that its address is 1600 Amphitheatre Parkway Mountain View CA 94043 USA, which is in Googleplex.

Host detection

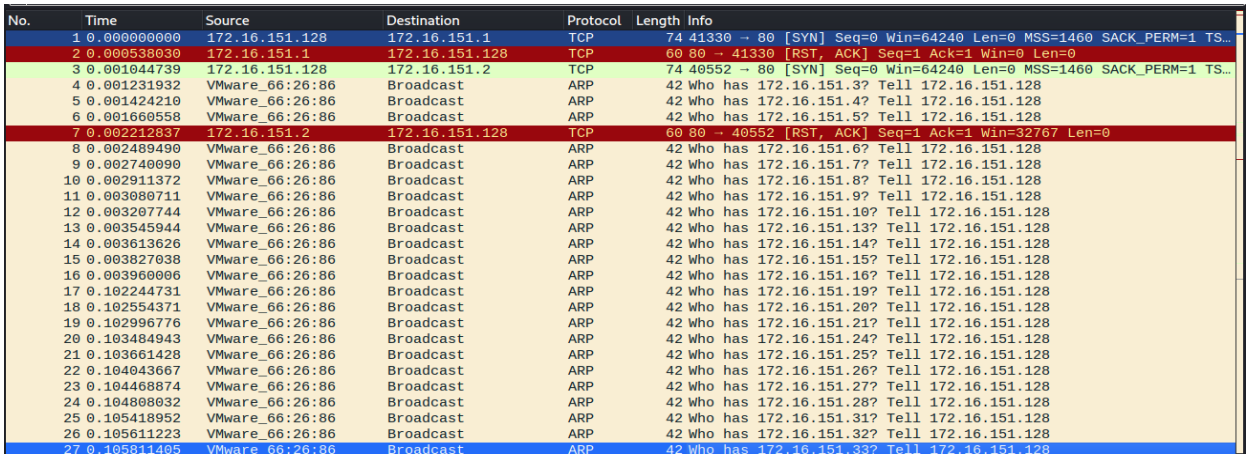
Local Network

IP addresses for active hosts in the local network and what entities they represent:

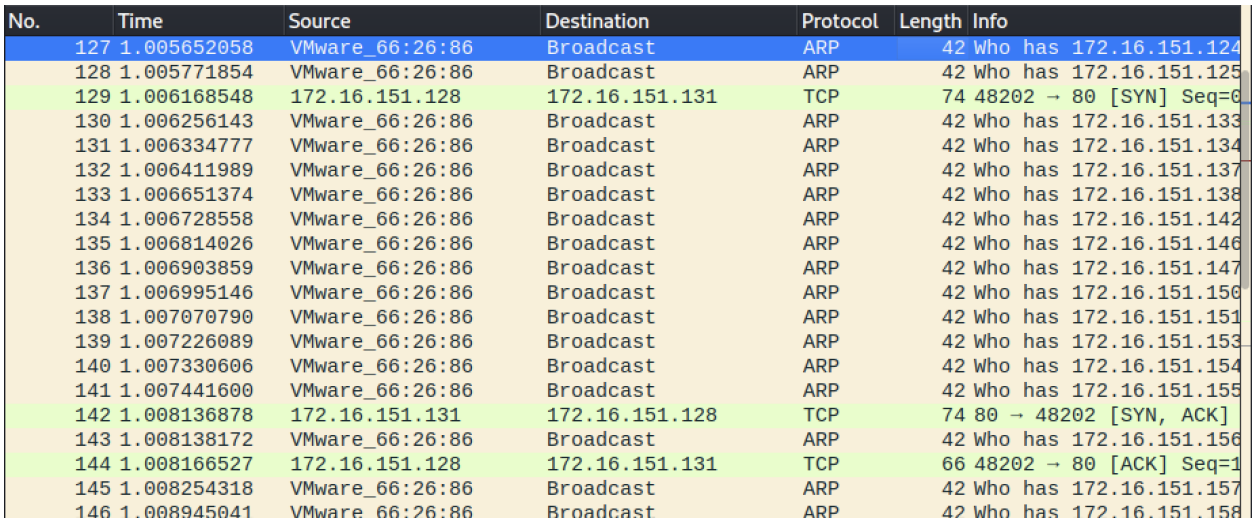
172.16.151.1(reserved IP address, belongs to Internet Assigned Numbers Authority)
172.16.151.2 (reserved IP address, belongs to Internet Assigned Numbers Authority)
172.16.151.128 - Kali
172.16.151.131 - Metasploitable

Steps:

For each IP address that nmap was searching in the local network, nmap sends PING scans. Kali first sends a [SYN] packet to check if the host is responding, then the destination will eventually send an [RST, ACK] packet if the port is closed, acknowledging and closing the connection; otherwise, it will send an [SYN, ACK] package indicating that the connection could be established. Wireshark shows a lot of attempts of ARP protocol on the destination Broadcast, which is a way our local network tries out all the IP addresses that start with the same 24 bits.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.16.151.128	172.16.151.1	TCP	74	41330 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TS…
2	0.000538030	172.16.151.1	172.16.151.128	TCP	60	80 → 41330 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
3	0.001044739	172.16.151.128	172.16.151.2	TCP	74	40552 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TS…
4	0.001231932	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.3? Tell 172.16.151.128
5	0.001424210	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.4? Tell 172.16.151.128
6	0.001660558	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.5? Tell 172.16.151.128
7	0.002212837	172.16.151.2	172.16.151.128	TCP	60	80 → 40552 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
8	0.002489490	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.6? Tell 172.16.151.128
9	0.002740090	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.7? Tell 172.16.151.128
10	0.002911372	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.8? Tell 172.16.151.128
11	0.003080711	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.9? Tell 172.16.151.128
12	0.003207744	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.10? Tell 172.16.151.128
13	0.003545944	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.13? Tell 172.16.151.128
14	0.003613626	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.14? Tell 172.16.151.128
15	0.003827038	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.15? Tell 172.16.151.128
16	0.003960006	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.16? Tell 172.16.151.128
17	0.102244731	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.19? Tell 172.16.151.128
18	0.102554371	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.20? Tell 172.16.151.128
19	0.102996776	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.21? Tell 172.16.151.128
20	0.103484943	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.24? Tell 172.16.151.128
21	0.103661428	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.25? Tell 172.16.151.128
22	0.104043667	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.26? Tell 172.16.151.128
23	0.104468874	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.27? Tell 172.16.151.128
24	0.104808032	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.28? Tell 172.16.151.128
25	0.105418952	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.31? Tell 172.16.151.128
26	0.105611223	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.32? Tell 172.16.151.128
27	0.105811405	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.33? Tell 172.16.151.128



No.	Time	Source	Destination	Protocol	Length	Info
127	1.005652058	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.124
128	1.005771854	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.125
129	1.006168548	172.16.151.128	172.16.151.131	TCP	74	48202 → 80 [SYN] Seq=0
130	1.006256143	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.133
131	1.006334777	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.134
132	1.006411989	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.137
133	1.006651374	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.138
134	1.006728558	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.142
135	1.006814026	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.146
136	1.006903859	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.147
137	1.006995146	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.150
138	1.007070790	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.151
139	1.007226089	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.153
140	1.007330606	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.154
141	1.007441600	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.155
142	1.008136878	172.16.151.131	172.16.151.128	TCP	74	80 → 48202 [SYN, ACK]
143	1.008138172	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.156
144	1.008166527	172.16.151.128	172.16.151.131	TCP	66	48202 → 80 [ACK] Seq=1
145	1.008254318	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.157
146	1.008945041	VMware_66:26:86	Broadcast	ARP	42	Who has 172.16.151.158

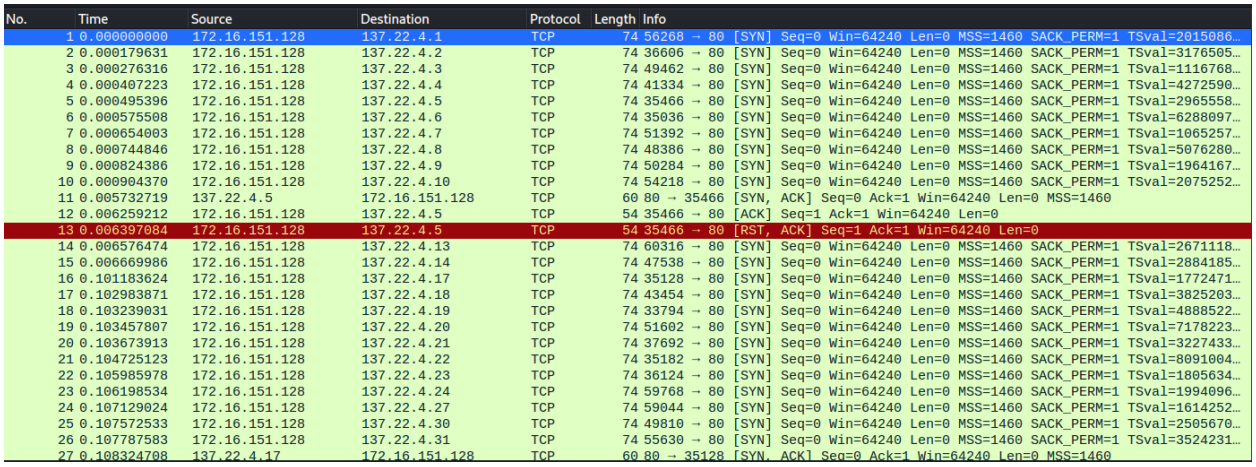
Network 137.22.4.0/22

IP addresses for active hosts in Math/CS department and what entities they represent (the entities are shown in front of each IP address i.e. 137.22.4.15 belongs to machine number 18 in olin 310):

elegit.mathcs.carleton.edu (137.22.4.5)
olin310-18.mathcs.carleton.edu (137.22.4.15)
perlman.mathcs.carleton.edu (137.22.4.17)
olin310-23.mathcs.carleton.edu (137.22.4.21)
olin302-01.mathcs.carleton.edu (137.22.4.30)
olin312-02.mathcs.carleton.edu (137.22.4.31)
olin304-07.mathcs.carleton.edu (137.22.4.32)
olin210cs70692.mathcs.carleton.edu (137.22.4.34)
olin210cs70686.mathcs.carleton.edu (137.22.4.35)
olin304-08.mathcs.carleton.edu (137.22.4.37)
olin304-06.mathcs.carleton.edu (137.22.4.38)
olin310-19.mathcs.carleton.edu (137.22.4.39)
olin310-17.mathcs.carleton.edu (137.22.4.40)
olin310-22.mathcs.carleton.edu (137.22.4.41)
olin210cs70687.mathcs.carleton.edu (137.22.4.42)
olin312-03.mathcs.carleton.edu (137.22.4.43)
olin210cs70691.mathcs.carleton.edu (137.22.4.46)
olin304-04.mathcs.carleton.edu (137.22.4.49)
olin310-21.mathcs.carleton.edu (137.22.4.54)
olin310-24.mathcs.carleton.edu (137.22.4.56)

olin310-20.mathcs.carleton.edu (137.22.4.57)
olin310-11.mathcs.carleton.edu (137.22.4.58)
olin310-15.mathcs.carleton.edu (137.22.4.59)
olin312-04.mathcs.carleton.edu (137.22.4.60
olin312-06.mathcs.carleton.edu (137.22.4.61)
olin310-12.mathcs.carleton.edu (137.22.4.63)
olin310-16.mathcs.carleton.edu (137.22.4.65)
olin304-03.mathcs.carleton.edu (137.22.4.66)
olin310-02.mathcs.carleton.edu (137.22.4.67
olin310-06.mathcs.carleton.edu (137.22.4.70)
olin310-03.mathcs.carleton.edu (137.22.4.71)
olin310-07.mathcs.carleton.edu (137.22.4.72)
olin310-04.mathcs.carleton.edu (137.22.4.73)
olin304-05.mathcs.carleton.edu (137.22.4.75)
olin312-05.mathcs.carleton.edu (137.22.4.77)
olin208-01.mathcs.carleton.edu (137.22.4.78)
olin310-08.mathcs.carleton.edu (137.22.4.79)
olin310-01.mathcs.carleton.edu (137.22.4.80)
olin310-05.mathcs.carleton.edu (137.22.4.82)
olin310-14.mathcs.carleton.edu (137.22.4.83)
olin310-10.mathcs.carleton.edu (137.22.4.85)
olin312-01.mathcs.carleton.edu (137.22.4.87)
olin310-09.mathcs.carleton.edu (137.22.4.88)
olin310-13.mathcs.carleton.edu (137.22.4.94)
olin310-is.mathcs.carleton.edu (137.22.4.95)
mmontee68381.mathcs.carleton.edu (137.22.4.98)
olin308-10.mathcs.carleton.edu (137.22.4.100)
olin208-02.mathcs.carleton.edu (137.22.4.102)
olin308-09.mathcs.carleton.edu (137.22.4.105)
olin304-09.mathcs.carleton.edu (137.22.4.106)
olin308-08.mathcs.carleton.edu (137.22.4.107)
olin210cs70693.mathcs.carleton.edu (137.22.4.110)
olin302-03.mathcs.carleton.edu (137.22.4.111)
olin308-07.mathcs.carleton.edu (137.22.4.112)
olin302-02.mathcs.carleton.edu (137.22.4.113)
olin304-01.mathcs.carleton.edu (137.22.4.115)
olin308-06.mathcs.carleton.edu (137.22.4.118)
olin308-02.mathcs.carleton.edu (137.22.4.121)
olin308-01.mathcs.carleton.edu (137.22.4.122)
olin308-04.mathcs.carleton.edu (137.22.4.123)
olin308-03.mathcs.carleton.edu (137.22.4.125)
olin308-05.mathcs.carleton.edu (137.22.4.127)
maize.mathcs.carleton.edu (137.22.4.131)
olin312-07.mathcs.carleton.edu (137.22.4.133)
olin321-62195.mathcs.carleton.edu (137.22.4.139)
wcc03168380.its.carleton.edu (137.22.4.141)
olin335-01.mathcs.carleton.edu (137.22.4.142)
olin319-62183.mathcs.carleton.edu (137.22.4.148)
olin327-62232.mathcs.carleton.edu (137.22.4.149)
olin339-62200.mathcs.carleton.edu (137.22.4.157)
olin304-02.mathcs.carleton.edu (137.22.4.188)
olin335-02.mathcs.carleton.edu (137.22.4.191)
t5.mathcs.carleton.edu (137.22.4.225)
mtietesting.mathcs.carleton.edu (137.22.4.234)

Steps:
Nmap tries to send ping scans. Instead of trying all of the possible IP addresses through the ARP protocol, it seems like the machine knows what IP addresses it should send the ping to. The rest is similar to when we did it on the local network, the destination addresses send back [RST, ACK] or [SYN, ACK] packets when they receive a [SYN] packet from kali, depending if the connection is closed or not.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.16.151.128	137.22.4.1	TCP	74	56268 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2015086…
2	0.000179631	172.16.151.128	137.22.4.2	TCP	74	36606 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3176505…
3	0.000276316	172.16.151.128	137.22.4.3	TCP	74	49462 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1116768…
4	0.000407223	172.16.151.128	137.22.4.4	TCP	74	41334 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4272590…
5	0.000495396	172.16.151.128	137.22.4.5	TCP	74	35466 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2965558…
6	0.000575508	172.16.151.128	137.22.4.6	TCP	74	35036 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=6288097…
7	0.000654003	172.16.151.128	137.22.4.7	TCP	74	51392 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1065257…
8	0.000744846	172.16.151.128	137.22.4.8	TCP	74	48386 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=5076280…
9	0.000824386	172.16.151.128	137.22.4.9	TCP	74	50284 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1964167…
10	0.000904370	172.16.151.128	137.22.4.10	TCP	74	54218 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2075252…
11	0.005732719	137.22.4.5	172.16.151.128	TCP	60	80 → 35466 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
12	0.006259212	172.16.151.128	137.22.4.5	TCP	54	35466 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
13	0.006397084	172.16.151.128	137.22.4.5	TCP	54	35466 → 80 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
14	0.006576474	172.16.151.128	137.22.4.13	TCP	74	60316 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2671118…
15	0.006669986	172.16.151.128	137.22.4.14	TCP	74	47538 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2884185…
16	0.101183624	172.16.151.128	137.22.4.17	TCP	74	35128 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1772471…
17	0.102983871	172.16.151.128	137.22.4.18	TCP	74	43454 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3825203…
18	0.103239031	172.16.151.128	137.22.4.19	TCP	74	33794 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4888522…
19	0.103457807	172.16.151.128	137.22.4.20	TCP	74	51602 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=7178223…
20	0.103673913	172.16.151.128	137.22.4.21	TCP	74	37692 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3227433…
21	0.104725123	172.16.151.128	137.22.4.22	TCP	74	35182 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=8091004…
22	0.105985978	172.16.151.128	137.22.4.23	TCP	74	36124 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1805634…
23	0.106198534	172.16.151.128	137.22.4.24	TCP	74	59768 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1994096…
24	0.107129024	172.16.151.128	137.22.4.27	TCP	74	59044 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1614252…
25	0.107572533	172.16.151.128	137.22.4.30	TCP	74	49810 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2505670…
26	0.107787583	172.16.151.128	137.22.4.31	TCP	74	55630 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3524231…
27	0.108324708	137.22.4.17	172.16.151.128	TCP	60	80 → 35128 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

Port scanning

- Which ports does Metasploitable have open, and what services do they correspond to (e.g. port 22 / SSH or port 80 / HTTP)?

  PORT STATE SERVICE
  21/tcp open ftp
  22/tcp open ssh
  23/tcp open telnet
  25/tcp open smtp
  53/tcp open domain
  80/tcp open http
  111/tcp open rpcbind
  139/tcp open netbios-ssn
  445/tcp open microsoft-ds
  512/tcp open exec
  513/tcp open login
  514/tcp open shell
  1099/tcp open rmiregistry
  1524/tcp open ingreslock
  2049/tcp open nfs
  2121/tcp open ccproxy-ftp
  3306/tcp open mysql
  5432/tcp open postgresql
  5900/tcp open vnc
  6000/tcp open X11
  6667/tcp open irc
  8009/tcp open ajp13
  8180/tcp open unknown

- What database server(s) is/are available on Metasploitable?

  mysql, and postgresql

- What is the value of the RSA SSH host key? What is the host key for?
  ssh-hostkey:
  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)

The host key is a unique value used to identify a computer, and its purpose is to authenticate computers and verify that the client is connecting to the correct host.

- Pick one of the open ports that has a service you have never heard of, and explain what the service does.
  - 445/tcp microsoft-ds: is a Windows port used by Server Message Block, which is a network protocol for sharing resources like files and printers over the network. This service is basically used every time you access a resource on Windows over the network.

References

https://security.stackexchange.com/questions/229820/microsoft-ds-vulnerability
https://hub.packtpub.com/discovering-network-hosts-with-tcp-syn-and-tcp-ack-ping-scans-in-nmaptutorial/