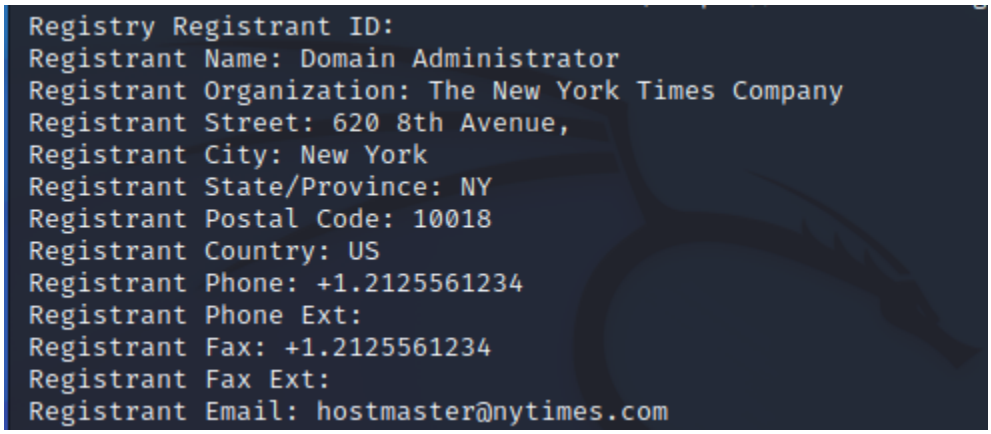
Pen Testing 1: Host Detection and Port Scanning
Author: Ashok Khare
Collaborators: Emma Roskopf, Lily Haas (did our own writeups)

1. Passive Information Gathering

- I investigated the domain ‘nytimes.com’
- Using nslookup, I found the domain’s IP address to be 151.101.193.164
- According to the data collected from the command ‘whois nytimes.com’, the Registry Expiry date is listed as January 19, 2023. Thus, this is the date on which the domain’s registration expires.
- The command ‘whois nytimes.com’ provided information for several individuals and organizations. The first was the domain registrant, whose name, organization, address, phone number, fax number, and email were provided. The specific details are:



- The second party the command provided information on was an administrator. Like the registrant, the admin’s name, organization, address, phone number, fax number, and email were provided by the command. Specifically:

- FInally, the third party the command provided information on was a technician. Like the previous two parties, the tech’s name, organization, address, phone number, fax

number, and email were provided. Specifically:

```
Registry Tech ID:
Tech Name: Domain Administrator
Tech Organization: The New York Times Company
Tech Street: 620 8th Avenue,
Tech City: New York
Tech State/Province: NY
Tech Postal Code: 10018
Tech Country: US
Tech Phone: +1.2125561234
Tech Phone Ext:
Tech Fax: +1.1231231234
Tech Fax Ext:
Tech Email: hostmaster@nytimes.com
```

- Finally, a little information was given about the domain management service that helped manage the domain. The Registrar's name, URL, WHOIS server, IANA ID, abuse contact phone, and abuse contact email were provided. Specifically:

```
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2022-02-11T17:23:54+0000
Creation Date: 1994-01-18T05:00:00+0000
Registrar Registration Expiration Date: 2023-01-19T05:00:00+0000
Registrar: MarkMonitor, Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895770
```

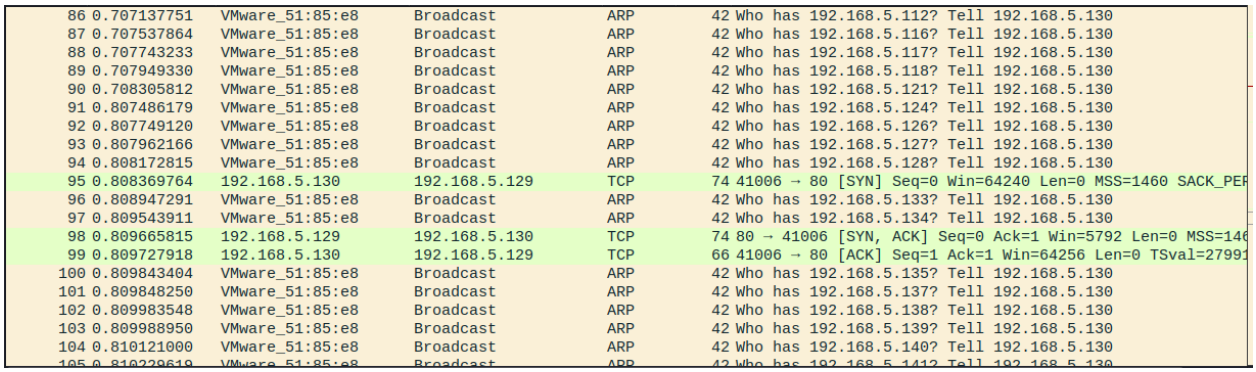
- According to Netcraft's information on the domain, the server is hosted somewhere in Western Europe. Also, the domain employs Google Analytics as a tracker.

2. Host Detection

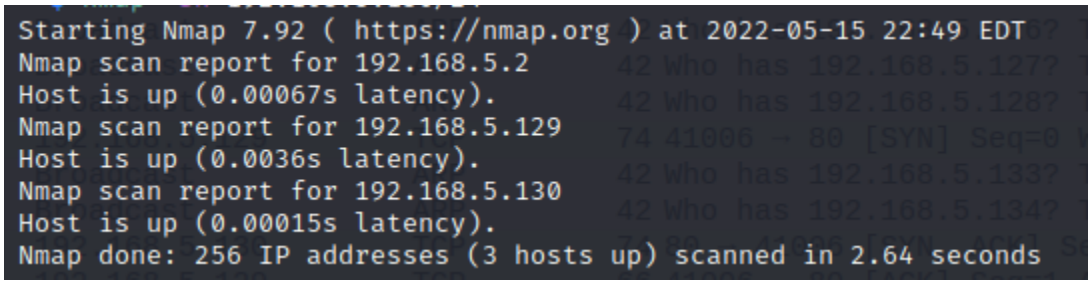
- The active hosts with 192.168.5.130/24 addresses on the local network were: 192.168.5.2, 192.168.5.129, and 192.168.5.130
- 192.168.5.130 is the origin machine (the one I used to run the query). There was no way to find the entities representing the other addresses using nslookup or ping.
- It appears that for most IP addresses in the specified collection on the local network, nmap sent ARP broadcasts to find the MAC addresses for those IPs. It did not send an ARP broadcast to its own IP address (because it already knows itself), but oddly, it did not send an ARP broadcast to 192.168.5.129 either. Another odd step that nmap took is sending TCP pings to a select few addresses: 192.168.5.1, 192.168.5.2, 192.168.5.129, 192.168.5.130, and 192.168.5.254. Nmap sent SYN packets to ports 80 and 443 at these addresses. Nmap received responses from its ARP broadcasts regarding the addresses 192.168.5.1, 192.168.5.2, and 192.168.5.254. However, the hosts nmap listed as open were 192.168.5.2, 192.168.5.129, and 192.168.5.130. Therefore, it seemed to me that nmap used an odd hybrid system of TCP pings and ARP broadcast responses to determine open ports. What is confusing is why nmap would send TCP pings to addresses on the local network and why it only listed some of the addresses it

got ARP or TCP responses from as open hosts (192.168.5.1 and 192.168.5.254 gave both ARP and TCP responses when reached out to, but neither was listed as open).

- Examples of addresses that did and did not receive the TCP SYN ping (note that the address that received and responded to the ping correlates to one of the open hosts in the summary):



86	0.707137751	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.112? Tell 192.168.5.130
87	0.707537864	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.116? Tell 192.168.5.130
88	0.707743233	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.117? Tell 192.168.5.130
89	0.707949330	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.118? Tell 192.168.5.130
90	0.708305812	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.121? Tell 192.168.5.130
91	0.807486179	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.124? Tell 192.168.5.130
92	0.807749120	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.126? Tell 192.168.5.130
93	0.807962166	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.127? Tell 192.168.5.130
94	0.808172815	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.128? Tell 192.168.5.130
95	0.808369764	192.168.5.130	192.168.5.129	TCP	74	41006 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PE
96	0.808947291	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.133? Tell 192.168.5.130
97	0.809543911	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.134? Tell 192.168.5.130
98	0.809665815	192.168.5.129	192.168.5.130	TCP	74	80 → 41006 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=146
99	0.809727918	192.168.5.130	192.168.5.129	TCP	66	41006 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=27991
100	0.809843404	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.135? Tell 192.168.5.130
101	0.809848250	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.137? Tell 192.168.5.130
102	0.809983548	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.138? Tell 192.168.5.130
103	0.809988950	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.139? Tell 192.168.5.130
104	0.810121000	VMware_51:85:e8	Broadcast	ARP	42	Who has 192.168.5.140? Tell 192.168.5.130



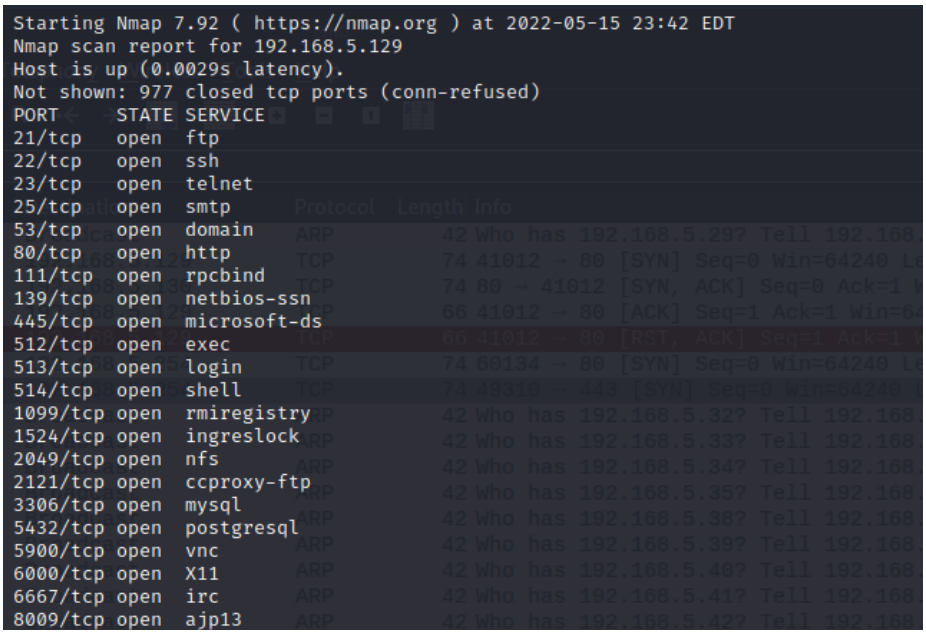
```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-15 22:49 EDT
Nmap scan report for 192.168.5.2
Host is up (0.00067s latency).
Nmap scan report for 192.168.5.129
Host is up (0.0036s latency).
Nmap scan report for 192.168.5.130
Host is up (0.00015s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.64 seconds
```

- The active hosts with 137.22.4.0/24 addresses on the Carleton network were: 137.22.4.5, 137.22.4.17, and 137.22.4.131
- 137.22.4.5 represents the Carleton Math/CS elegit server; 137.22.4.17 represents the Carleton Math/CS perlman server; and 137.22.4.131 represents the Carleton Math/CS maize server
- For each possible candidate IP address, because the address was not on the local network, nmap tried to establish a TCP connection with ports 80 and 443 (HTTP and HTTPS) at the address by sending a ping. nmap sent a TCP SYN packet to each address; addresses that responded to the ping were the addresses of the open hosts. Addresses that didn't respond did not correspond to an open host. However, an odd aspect of this method is because nmap only pinged ports 80 and 443, it wouldn't have picked up open hosts that didn't have web servers running. Therefore, there could have well been more open hosts than nmap found on the Carleton network.

3. Port Scanning

- Addresses and open ports on Metasploitable
    - The address corresponding to Metasploitable is likely 192.168.5.129, as it has 23 open ports. This is in the context of 192.168.5.2, which had one open port, and 192.168.5.130, which had no open ports. The relative abundance of open ports for the supposed Metasploitable address suggests that it indeed corresponds to Metasploitable.

- The open ports on Metasploitable, along with the corresponding services, are as follows:



  - Another open port (not pictured in the above screenshot) is port 8180, an unknown open tcp port
- Available database servers on Metasploitable
  - I determined which Metasploitable ports were likely to involve database servers by looking at the Wikipedia articles for each port specifically, as well as the Wikipedia article linked to the assignment about common ports.
  - Database servers available on Metasploitable include domain (port 53), mysql (port 3306), postgresql (port 5432), microsoft-ds (at least the Active Directory section) (port 445), rmiregistry (port 1099), ingreslock (1524) (ingreslock technically just interacts with the ingres database, but since it grants access I'm counting it)
- RSA SSH host key information
  - The RSA SSH host key seems to be the host's public key, which can be used as a means of authentication and could be stored by the client to recognize the host later on. Information found at https://www.ssh.com/academy/ssh/host-key
  - The particular value for the RSA SSH key for the Metasploitable host is 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3
- Information about ingreslock service:
  - The ingreslock service is used to lock parts of an ingres database
  - Locking an Ingres database prevents a process from altering or processing particular pieces of data, according to https://www.dbta.com/Columns/DBA-Corner/Understanding-Database-Lock-Timeouts-and-Deadlocks-148659.aspx
  - According to its Wikipedia article, the Ingres database is an "SQL relational database management system." It is designed to manage multiple databases, and support applications used by governments and large corporations.

- The ingreslock port can be used as a backdoor to another system if it is left open when not needed, according to http://www.rwbnetsec.com/ingreslock/ (and all the other sources on the first page of Google when you search ‘ingreslock’)