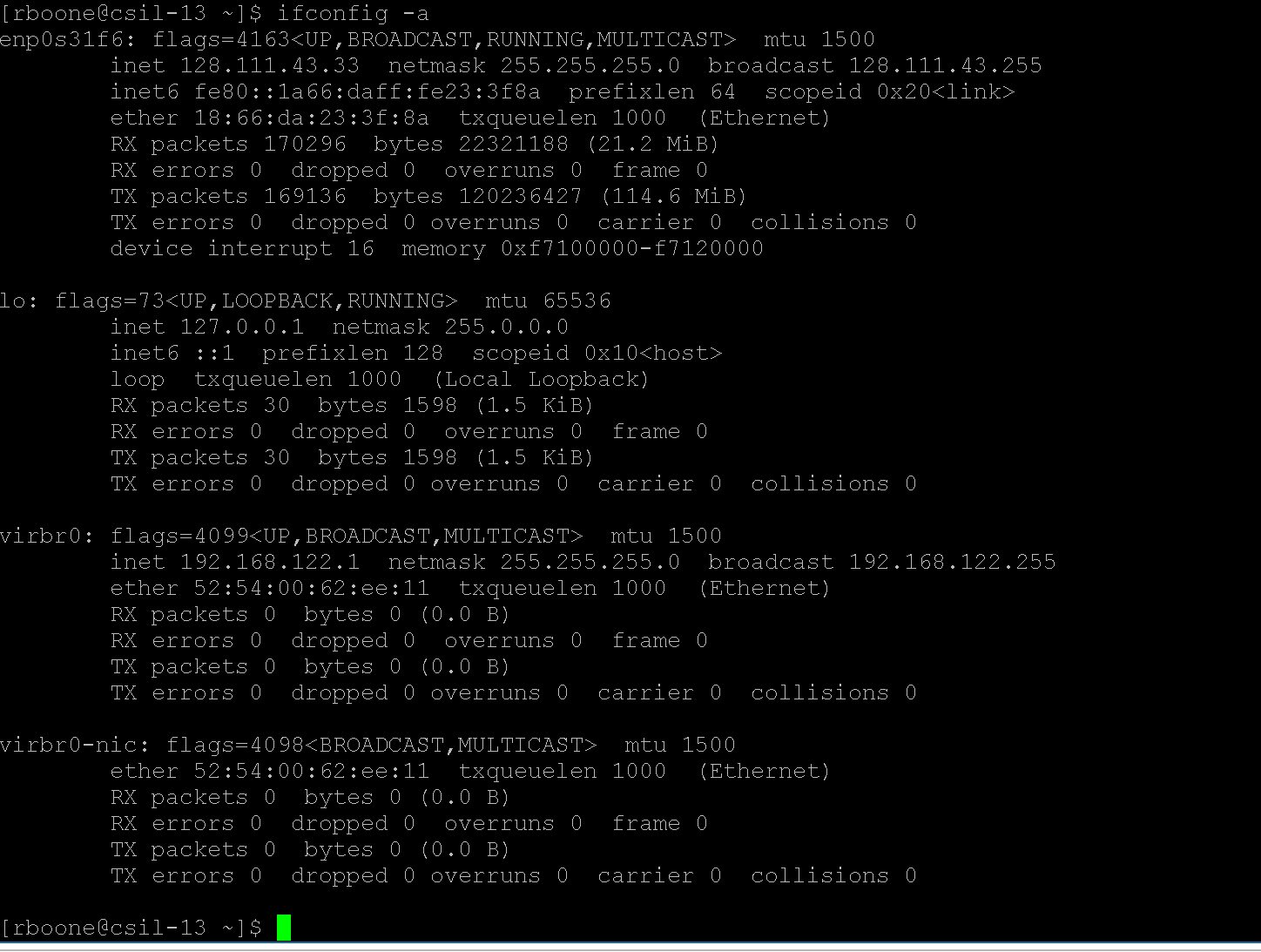
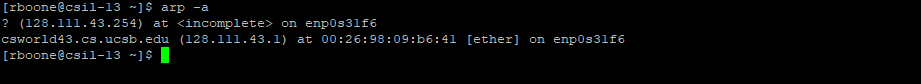
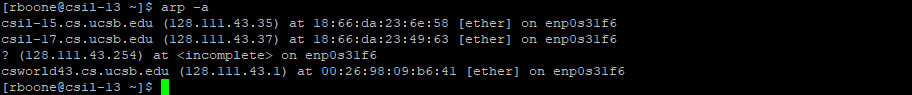
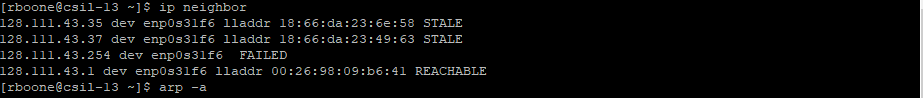
CS 176B HW1

1. 
   1. The network interface is an ethernet interface listed as enp0s31f6. It gives it’s own IP address (128.111.43.33), and the netmask and broadcast address for the computer. On the next line, it shows the IPV6 address with a prefixlength of 64, indicating that it’s using the whole ipv6 address. The next line indicates the ethernet address on the local ethernet connection with the length of the transmit queue set to 1000. The next four lines indicate the packets and bytes sent and received, as well as the amount of errors and dropped packets. The most significant things to be noticed here are that the TX and RX packet numbers are very similar, but the computer is receiving many more bytes than it is transmitting. Additionally, there are no errors or dropped packets, indicating that we have a very stable connection. After the end of this interface, we see the other interfaces.
   2. The same (or very similar) output can be generated with the command “ip addr” on most linux machines.
2. ARP
   1. 

The first line indicates that my machine know’s there’s a machine at 128.111.43.254 but does not know the hardware address for said machine. The second line indicates that the machine with url “csworld43.cs.ucsb.edu is at IP address 128.111.43.1 and hardware address 00:26:98:09:b6:41 accessed through an ethernet connection.

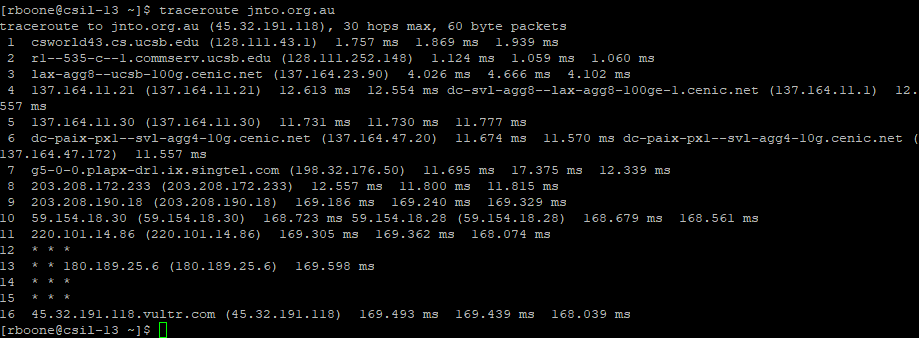
* 1. When trying to delete pieces of the ARP table I get the error “SIOCDARP(dontpub): Operation not permitted” and when trying to add I get the error “SIOCSARP: Operation not permitted”. Both these indicate that I do not have permissions to edit the ARP table. This makes sense because editing the ARP table can cause large security flaws.
  2. 

You can affect the ARP table by connecting to other hosts that will be accessible to you by hardware address. In this case, I connected to csil-15 and csil-17 by using the ping command.

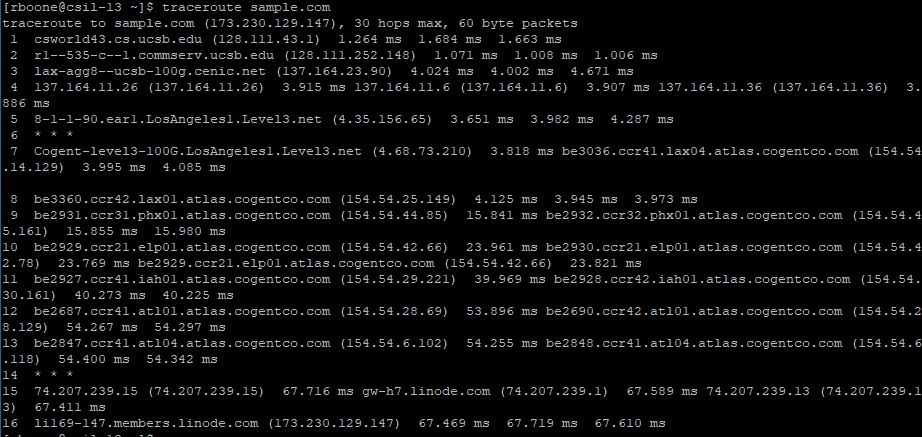
* 1. The simplest method to discover a timeout value is just to add the value in (either manually or by pinging as above) and then check the ARP cache with “ip neighbor” every few seconds or so to get an estimate. The default timeout can also be found by using the command “cat /proc/sys/net/ipv4/neigh/default/gc\_stale\_time”. In my case, it’s 60 seconds which fits well with the time at which ip neighbor marked the entry as “stale”. However, the cache numbers do not disappear from the arp -a table. The reason for this is unclear.
  2. Ip neighbor indicates which arp cache entries have been refreshed recently enough to be good and which are old enough that they should not be used. 

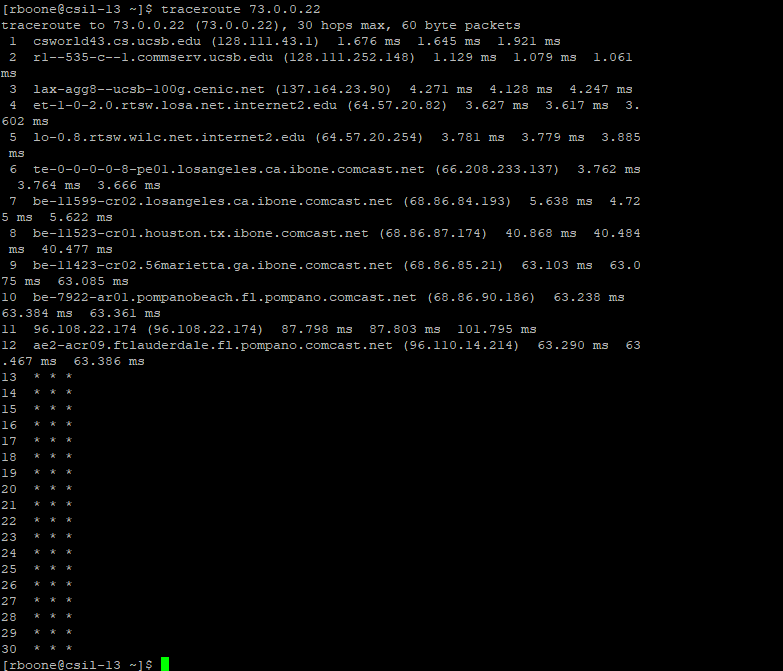
As you can see in the screenshot above, the connections I had previously pinged are not listed as “STALE” because they have not been used recently.

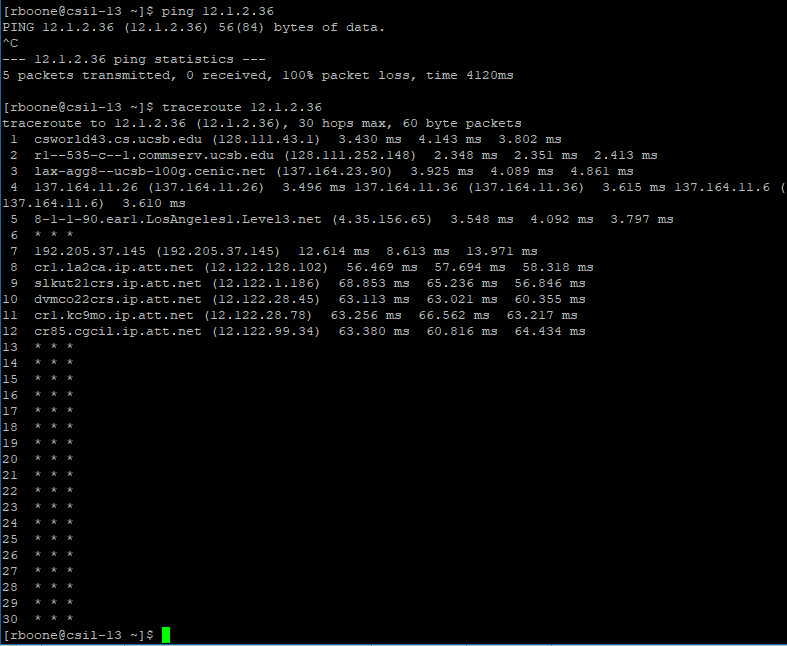
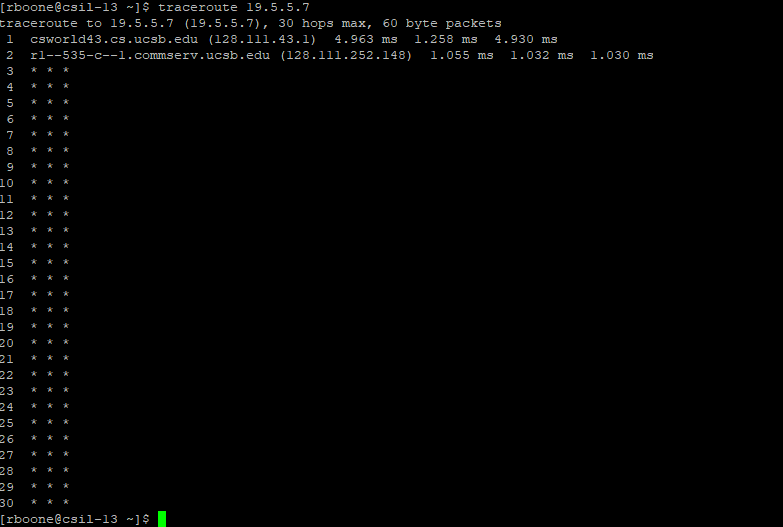
* 1. If the two hosts are not in the same subnet, likely no problems will occur. If the two hosts are in the same subnet, it will cause huge problems. Traffic going to either one of the two hosts will randomly go to one of them depending on whose address was last recognized by the switch. Neither will be able to effectively get an internet connection because packets will consistently be sent to the wrong machine.

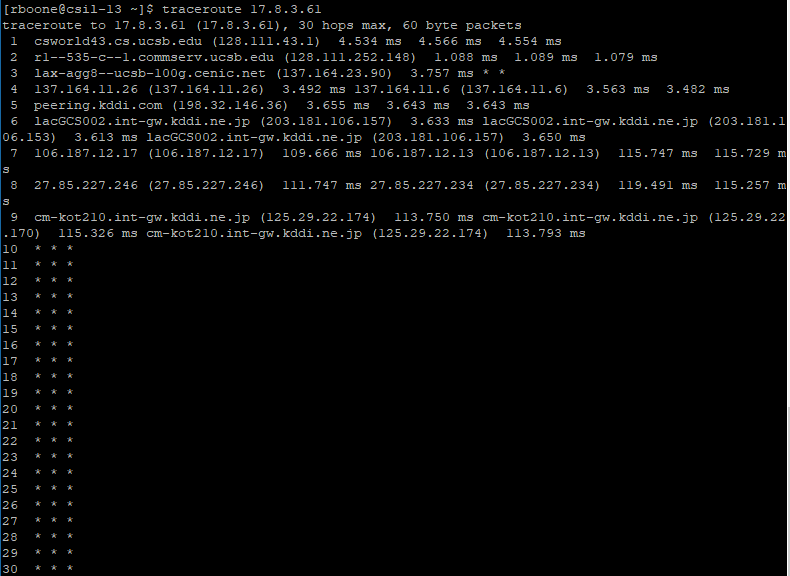
1. Traceroute
   1. Traceroute maps out a path to a location by sending packets with increasing TTL values, starting at a TTL value of 1 to get the closest path. In Linux, these packets are automatically UDP packets although this can be changed. When a router receives a packet, it automatically decrements its TTL value by 1 and if the TTL value is decremented to 0, the router discards the packet and (usually) return an ICMP Time Exceeded packet to the source. When the machine running the traceroute receives the ICMP Time Exceeded packet, it measures the time from sending the packet to receiving it and displays this time. Usually, 3 packets of each TTL value are sent and measured until traceroute gets values for the destination IP address.
   2. 

The above shows the path traceroute found from csil-13 to jnto.org.au. Lines 1-3 show the path the packets took out of the UCSB network. Lines 4-11 show the continuous path of the packets on their way to jnto.org.au. Notable are lines 4, 6, and 10 where some of the packets return different IP addresses indicating that they took different paths and did not all end at the same router. Lines 12-15 show mostly stars, indicating that traceroute did not receive the ICMP Time Exceeded packet for these numbers. This is usually because the router that received them did not send any packet back. At line 16, we reach jnto.org.au. However, it is not listed as jnto.org.au but as 45.32.191.118.vultr.com because this is the reverse DNS lookup address our computer has found.

* 1. 
  2. Machines can be configured to not respond to pings, so a lack of response to pinging a machine doesn’t tell you anything about whether the machine exists or not. If we traceroute the machine and get all stars, or continuous stars after a point, we can know the machine does not exist.
  3. I chose to use 73.0.0.22. I chose to start with the comcast IP range 73.0.0.0/8 and scan it using Angry IP Scanner. It found that many IPs that were inaccessible, so I chose one at random and tested it on my own machine using ping and traceroute. The traceroute was as follows.

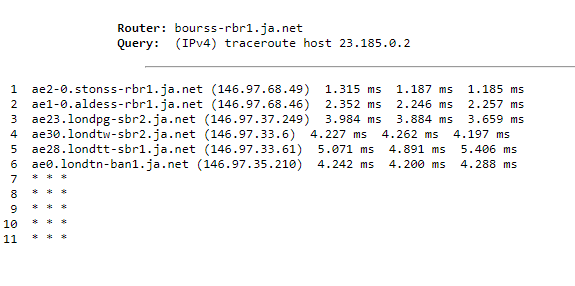


* 1. 
  2. 

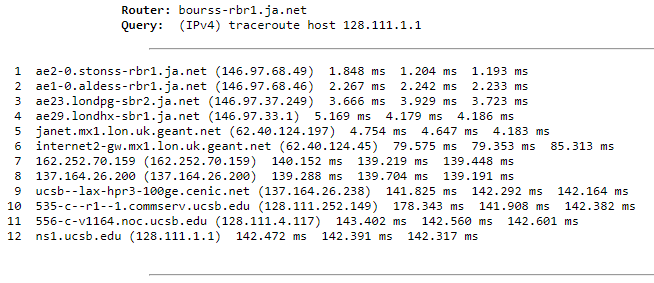


The routes all go some distance, then turn into complete stars. This happens because it is impossible for all the individual routers to know whether a machine exists, so each of the routers does their best to direct the packet towards its intended destination. At some point (likely when the packet reaches whatever subnet the computer should be contained in, the packets just get dropped because the host does not exist.

* 1. Tracepath is very similar to traceroute, but it does not show the IP of each individual router. It just shows the router name and a single TTL value.

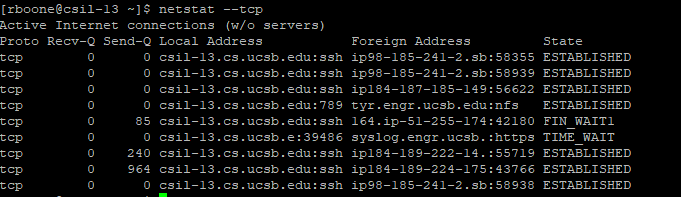
1. 1. 

All of the routers are within the ja.net domain, just like alice.ja.net. Line 1’s location is not clear. Line 2 routes through Aldess in France. Lines 3-6 all route through London. Although the exact locations are not clear it is likely that the router for line 4 is in twickenham, and the router for line 6 is in Tunbridge. Because the route ends very early, and the IP address is not a UCSB IP address, it is clear that ucsb.net is not hosted in UCSB.

* 1. 

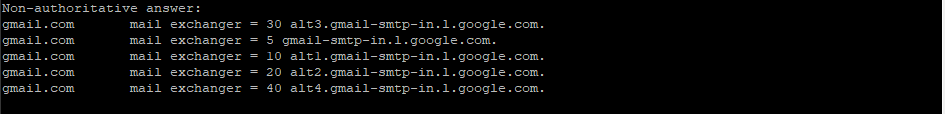
Unlike the last traceroute, this route actually finishes, and travels all the way to UCSB. Just as before, the first four lines are within ja.net with all four routers being the same as before. Lines 5 and 6 travel through London. Lines 7 and 8 are unclear because they only offer IP addresses. Lines 9 to 12 show the path of the packets through the UCSB network to the given machine.

1. 1. Netstat is a linux command that gives information about the network connections on your local machine. It is used to monitor network activity on a machine.
   2. Netstat --tcp will show all tcp connections. The connections on the machine I used (csil-13 look as follows:

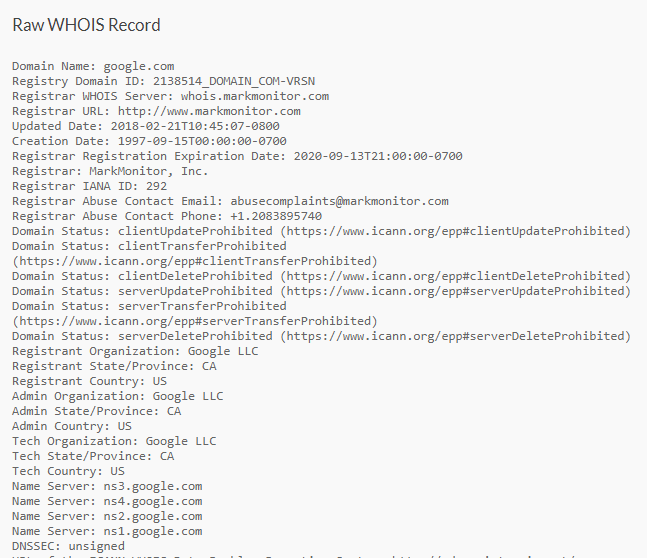


* 1. Netstat -I displays interface status. There are 3 interfaces on my machine (csil-13). The loopback interface allows the machine to network with itself. Anything sent to this interface will loop back to the local machine. This can be used as an easy way to test web servers and other internet traffic without opening them to the public. The loopback interface is the same as using IPV4 address 127.0.0.0.

1. nslookup
   1. Cs.princeton.edu is at 128.112.136.51
   2. My computer uses 128.111.1.2. The computer is told by the local network what DNS server to use. Because both the host (csil-13) and the DNS server (128.111.1.1) are on the UCSB network, the local router probably gave this DNS address to the host on connection to the network.
   3. Nslookup -type=mx hotmail.com gives the name of the mail exchanger: hotmail-com.olc.protection.outlook.com. By pinging this URL, we can get the IP 104.47.45.33.
2. whois
   1. Kevin Schmidt is the administrative contact for UCSB.
   2. 130.207.8.11 is the ip address for mortician.cc.gateway.edu
   3. Call Mark Silis at 617-324-5900
   4. Running nslookup -type=mx gmail.com gives the following mail exchangers:



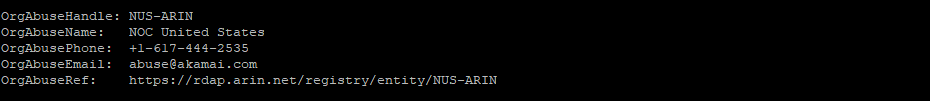
Checking this in a WhoiS database gives the following information.



This shows that the gmail network is coordinated by MarkMonitor, a digital brand protection company.

* 1. Innovative Logic Corp owns 198.182.196.56. Arin.net acts as the dns nameserver. 198.182.196.0 acts as DNS nameserver.

1. Misc
   1. 104.69.73.91 is returned. The browser returns this as an “invalid url”
   2. I am redirected to [www.whitehouse.gov](http://www.whitehouse.gov). [www.whitehouse.gov](http://www.whitehouse.gov) is at a different ip, and so is a different website. This means that the computer that hosts 104.69.73.91 is set up to redirect people who access by URL but not to redirect people who try to access it by IP.
   3. Abuse complaints should be sent using any of the following information:



The domain is not a .gov, but a .com indicating that abuse complaints are received outside the government.

* 1. Abuse complaints should be sent to [registrar-abuse@akamai.com](mailto:registrar-abuse@akamai.com). This is the same domain as above, once again indicating that abuse complaints are received outside the government.