**Spoofing MAC Address and Deny Internet Access to a Computer**

Arjun Sethi

Keith Nguyen

**Abstract**

Each computer connected to the internet must have both and IP address and a MAC address. The router maintains an Address Resolution Protocol (ARP) table that pairs IP and MAC together. At the moment, we know that MAC address cannot be the same for computers in a network and each computer also keeps its own ARP table. Therefore, we want to know what happens if two computers in a network have the same MAC. And what happens if two computers have the same IP. Finally, we want to see if the server is confused between two computers having the same MAC address or the same IP address, how do we listen or pretend to be one of the computers by changing our IP or MAC address.

Finally, if the MAC address is mutually exclusive, we look for a way to disconnect the real computer from the network long enough to get in its place, or to become the middleman between the router and the real computer. We explore possibilities such as DoS attack, physical blocking and other methods.

**Concepts**

* MAC address
* ARP table
* IP address
* DoS attack
* Physical blocking

**Spoofing MAC Address**

We use Ubuntu and CentOS as our preferred operating system as Linux gives us more control over a computer’s lower level data. They also have many penetration testing tools from Kali Linux so we have more options to explore our project. There are several ways to change the MAC address on Linux: use built in “Edit Connection”, use macchanger, use ifconfig.

* sudo ifconfig <interface> down
* sudo ifconfig <interface> hw ether <new MAC>
* sudo ifconfig <interface> up

On Macchanger, we replace the middle command with macchanger:

* sudo ifconfig <interface> down
* sudo macchanger -m <new MAC> <interface>
* sudo ifconfig <interface> up

To get the ARP table of our computer, we use

* ip route
* arp -a

To change IP address, we use ifconfig:

sudo ifconfig <interface> <new IP> netmask 255.255.255.0

**Success test 1:**

1. Computers A,B,C connect to network
2. A sends message to B, B receives message
3. C assigns itself B mac address and reconnects to network
4. Router gets confused, and disconnects B from network
5. Router assigns C , B’s ip address that was associated with B’s mac address by looking this up in the local arp table
6. C sends a gratuitous ARP to A to let him think that B is still online
   1. Since C mac address is the same as B, A does not make any updates to its arp table and does not realize that C has B’s identity
7. A sends message to B, but C receives it

**Security issue found:**

However, it should be noted that if B connects, speaks to A, and then leaves the network, arp tables are not updated appropriately. C can use B’s mac, will be assigned B’s old ip, and A will can start communicating with C thinking it is B. C can leave the network. B can come back online, and start back where it left off. A will have no idea it sent packets to someone else, B will have no idea its identiy was stolen, and worst part is C is hiding in plain sight and there is zero trace of who C actually was because his real mac was never noted. This also means that on a network, C can take any person’s identity that was previously on the network, send and receive, and then leave without being noticed.

**Problem with this:** depending on the router, we noticed that B and C will continue to get kicked off the network back and forth in step 4. If B gets kicked off, then this is good because C is able to talk to A without A knowing, and B isn’t able to tell A that it is kicked off the network because it can’t even connect to the network unless it changes its mac address. If C gets kicked off, then this is not good because our attack is unsuccessful

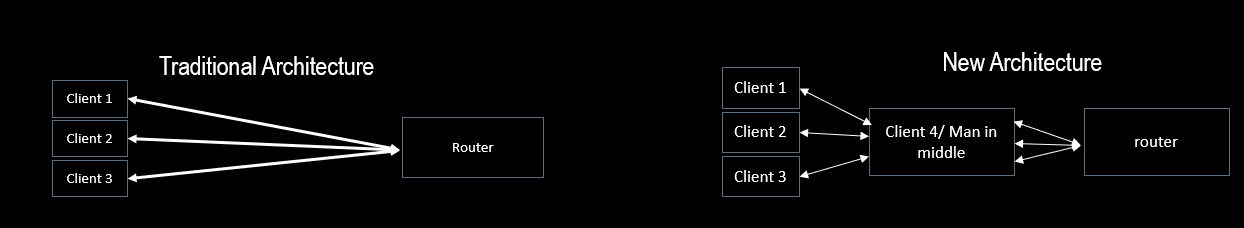
**Success test 2:**

1. Computers A,B,C connect to network
2. A sends message to B, B receives message
3. C takes B’s ip address
4. C sends router gratious arp that he has B’s ip address
5. Router updates arp table to reflect C has B’s ip address
6. B gets a new ip address assigned to it
7. A sends message to B’s old ip address, but C receives it
8. A’s arp table is updated with new mac address

Problem with this: A will know mac address changed even though ip stayed the same. Depending on security model, A will know something has gone wrong

**Future test:**

1. Try to have C take router’s ip address so it is now looks like the “gateway” to the internet. Then, any packets sent or received from the router or clients goes through C. If C looks like the router, the clients shouldn’t be able to tell that they are talking to someone else. The router will know, but since it is automated and security is terrible on routers, it might not even care and just allow everything to pass through that middle man.



**Discovery**

We discovered that MAC address cannot collide in the same network. However, using this characteristic of the router, we can perform a Denial of Service attack on smaller computers that do not connect to the network often.

Using the same principle, if we have a reliable way to cut internet connection to the target computer, we can use that period of time to listen to the data transmission from router.

Another use of this is to temporarily remove access from the computer to the network and become the middle man.