

1. If node1 is a "man in the middle" then node4 is an "odd man out." In particular, node 4 was unaccounted for in section 3 "Recording actual address mappings." Later you arp poisoned node2 and node0 from node1; how about arp poisoning node 4 from node1? You accomplish poisoning by sending a crafted arp message to a node. Comment on the ways and means of poisoning node4 from node1.

1. node4 is basically out of the network, all on its packets must go through node0. So if we poison arp cache of node0 from node1 (which we can) for all valid ip addresses in the subnet of 10.1.1.0, all packets which node4 can possibly send to any host in the 10.1.1.0 subnet are intercepted by node1. This is how we can indirectly poison node4.

2. Answer the question at the end of section 6. Under the circumstances of that section, "How does traffic between node2 and node0 get from node2 to node0?"

Any packet sent by node2 to node0 goes like this:

node2 sends a packet to the hardware address corresponding to node0’s ip address. This is node1’s hardware address.

Node1 after receiving this packet sends this packet to node0’s hardware address.

Thus the packet goes like this node2 -> node1 followed by node1 -> node0

3. Answer the question at the end of section 7, "How?" Recall that node2 logged into ftp on node4 and somehow node1 figured out the user password given by node2. How?

node1 was able to figure out the username and password given by node2 to node4 because it has poisoned the arp cache of both node2 and node0. Node1 is basically a man in the middle and intercepts, examines/modifies and forwards all packets flowing between node0 and node2. All packets to node4 must flow through node2 since that is the only connection node4 has to the network. During the ftp login process node2 sent a packet containing the username and password to node0’s ip address which then forwarded this packet to node4. However the hardware address to which node2 sends the packet to belongs to node1 and not node0 (because of cache poisoning). Since the username and password were sent as cleartext and node0 was filtering all packets it receives for these keywords, it was able to intercept and recover them.

4. Imagine you run a web hosting company. The manager at one of your clients, a medium sized business, calls you in alarm and reports the apparent defacement of his website running on your host machine. Images on the site have all been replaced with various hacker images like the laughing skull. He heard about it from several of his employees, then saw it with his own eyes on their terminals. His website has fallen victim to the same mischief as the one on our node4. What is your course of action?

In such a scenario one of the quick fixes would be to completely flush the existing entries in the arp cache and rebuild it. This would correct the entries in the arp cache and should restore the correct images on the website. To prevent someone from remounting an arp poisoning attack we could disable processing of incoming arp replies i.e only process arp reply messages which correspond to outstanding arp requests.

We could also try moving the web server to the edge of the network so that a man in the middle arp attack would not be physically possible. i.e incoming requests are directly handled by the web server and are not passed through other hosts on the LAN.