COMP 8505 – DNS Spoof (Assignment 3)

Andrew Maledy

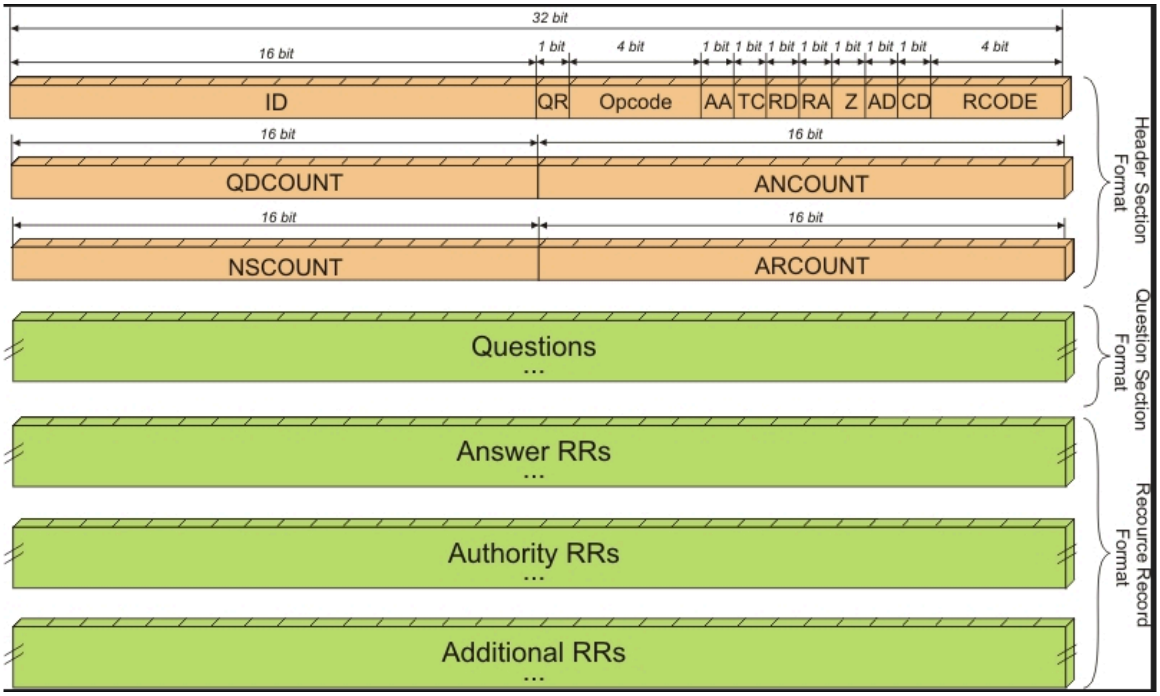
May 29, 2013

Design

This application will listen for DNS requests from a *specified host* and respond with a spoofed query, effectively initiating a man in the middle attack. *Specified host* means that this application does not compromise an entire network, merely a single host on the network. This greatly reduces the risk of being caught as we can pinpoint one target and take them down accordingly.

For the ARP poisoning part of this assignment, we'll be using PacketFu's built-in library for ARP packets.

Knowledge of the DNS protocol is needed in order to begin development of this application. Examine the following figure:



There are several key fields we'll be exploiting in our application. The first, the identification field is needed because clients are looking for a matching ID in the DNS server's response. The QR field indicates whether or not the packet is a DNS query or a DNS response. Thirdly, we'll be analyzing the question field inside incoming packets to determine which domain name the user is asking about. Finally, we'll craft a packet and set the Answer Resource Record field with our IP address and send the packet to the client.

Psuedo-code

1. Spawn Thread 1 (listen for ARP Requests)

- When ARP request is received from a specified sender– respond to sender with middle man's MAC address instead of gateway address.

2. Thread 1 is entirely dedicated to maintaining poisoned client's arp tables. At no point does it stop responding to ARP requests. As long as Thread 1

Spawn Thread 2(listen for UDP on port 53 (DNS) traffic from victim)

- When a DNS packet comes in, determine whether or not it's DNS.
    - The first 16 bits are the Identification, the next 2 bits are whether or not the packet is a query or response. (10 for query). If the UDP payload in that bitplacement is 10 we know it's a DNS request.
- Next, extract the ID from the packet. We need to use that in the response.
- Thirdly, get the DNS hostname that's being requested. This is complicated but with knowledge of the protocol's formatting syntax it won't be *too bad.*
- Craft a response packet and return to sender.

3. When CRL+C is captured, stop both threads.

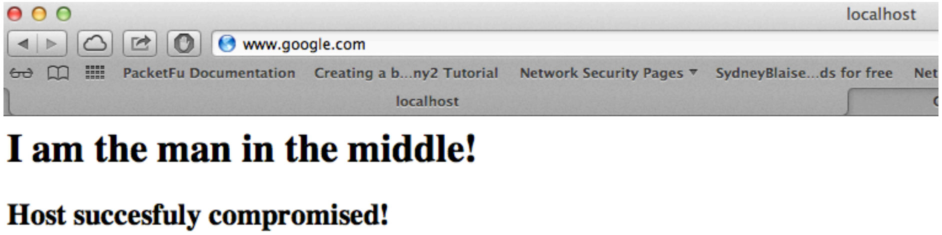
Testing

Man in the middle browser test (passed)

For this test, our victim host will attempt to reach www.google.com. Our man in the middle, will capture the DNS query, and send back a fake response packet.

Success case: The spoofed web server's page appears when the client tries to reach www.google.com.
Result: Pass

Evidence:



I am the man in the middle!

Host succesfuly compromised!

Above you can see that our victim attempted to reach www.google.com and in fact reached our victim's local webserver. Our attack has been successfully carried out. It should be noted that in production you would want to display a page that looks identical to Google to avoid detection. Facebook or a banking website may be more useful however these activities are not condoned by this report and are *highly illegal*.

Spoofed DNS packet vs actual DNS packet (passed)

Here we'll analyze a legitimate DNS packet vs our crafted DNS packet to determine whether or not our crafted packets are of the correct syntax.