Fiddling with DNS

# BACKGROUND

Recently, an [article was published](https://news.ucr.edu/articles/2019/04/03/computer-scientist-beats-hackers-their-own-game) that featured a doctoral student named Fatemah Alharbi who successfully perpetrated a DNS hack while studying security.

DNS (domain name service) is a worldwide service that converts URLs into IP addresses (so when you type “www.google.com”, a DNS server will convert that into the proper IP (172.217.10.78).

Computers have a few different ways to access DNS. It used to be the case that every computer had a file on it called “hosts”, and that would map every computer in the world to a static IP address. That process became impractical as the number of computers proliferated, as everyone would have to update their “hosts” file every time a new computer became connected to the internet! Surprisingly, that’s still the first place computers go to when trying to resolve a domain. This is true in Windows (c:\Windows\System32\Drivers\etc\hosts), MacOS (/etc/hosts) and Linux (/etc/hosts).

Since most hosts files do not have many entries, almost all DNS queries are retrieved from the DNS cache on the computer, the DNS cache of the ISP, or the DNS entries in a server. But it’s important to note that the hosts file is really the first stop!

DNS lookup is an amazing process--much more detailed than we will go into in this class--but there are a few things to highlight from a security standpoint. Most DNS attacks are done at a DNS server level--DNS cache poisoning. This [new attack](https://news.ucr.edu/articles/2019/04/03/computer-scientist-beats-hackers-their-own-game) exploited the fact that the DNS resolution (the answer to “What is the IP address of www.google.com?”) is transmitted in plaintext, so it is fairly easy to intercept and spoof.

# DESCRIPTION

What we will be doing today is working through how to change the hosts file entries to demonstrate what a DNS resolution attack might look like. Researchers have successfully [changed DNS lookups on exposed routers](https://arstechnica.com/information-technology/2019/04/ongoing-dns-hijackings-target-unpatched-consumer-routers/), too (thanks, [Shodan](https://www.shodan.io)!). So make sure you lock your router down!

# REQUIREMENTS

A web browser, a Linux install (VM), and an internet connection.

# PART I: Ping a known website

1. To establish a baseline, we will first send some ping requests to a known website:

**ping -c 4 google.com**

That will send four pings (where -c means “count” and “4” is the number of counts) to google.com and we can see the IP address.

| EVIDENCE #1 |
| --- |
| **PASTE THE IMAGE OF THE ping CALL** |

# PART II: Modify the hosts file

1. Edit the hosts file (found in /etc):

**nano /etc/hosts**



1. Append a line at the end that will override a query to google.com (and www.google.com) to this IP:

**5.9.243.187 google.com www.google.com**

| EVIDENCE #2 |
| --- |
| **PASTE THE IMAGE OF THE CONTENTS OF THE hosts FILE AFTER MODIFYING IT** |

# PART III: See the results in your browser

1. We can’t always rely on Firefox or Chrome to look at the hosts file first during DNS resolution, but we *can* use a text-based web browser to test this. Install lynx if it isn’t installed:

**sudo apt update
sudo apt install lynx**

1. Use it to browse the internet! Try going to google.com:

**lynx google.com**

You might have to accept/reject cookies.

You may have to accept the invalid certificate by pressing the ‘y’ key when prompted (normally this is not advisable, however, the certificate will necessarily NOT match google.com’s certificate)

| EVIDENCE #3 |
| --- |
| **PASTE THE IMAGE OF THE lynx CALL AFTER FIDDLING[MAKE SURE THE NAME OF THE WEBSITE IS VISIBLE - HINT: IT IS NOT GOOGLE]** |

1. Go back into the hosts file and remove the portion you added so your computer will behave appropriately *and test the DNS.*

**nano /etc/hosts
ping -c 4 www.google.com**



1. Recognize that modifying hosts file is not a popular attack (though it is probably effective because most users do not know how DNS resolution works). For it to happen, attackers need to have access to your computer.

The point of this exercise, though, is to demonstrate how DNS resolution can be compromised and is one of the vectors that can be exploited. Had this been an intentional attack:

* The redirect would not be benign
* It probably would not have happened at the hosts file level (it would most likely be a consequence of DNS cache poisoning)

Note that changing the hosts file may not affect how browsers (like Firefox or Chrome) resolve domain names; that is, they may bypass the hosts file because of technologies like predictive browsing. You can disable these features in the browser.