

[irongeek.com](https://www.irongeek.com)

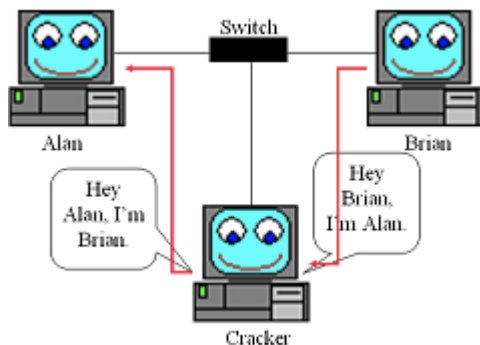
Basics of Arpspoofing/Arppoisoning

*Irongeek.com*6-7 minutes

The Basics of Arpspoofing/Arppoisoning

ARP stands for Address Resolution Protocol and it allows the network to translate IP addresses into MAC addresses. Basically, ARP works like this: When one host using IP on a LAN is trying to contact another it needs the MAC address (aka: hardware address) of the host it is trying to contact. It first looks in it's ARP cache (to see your ARP cache in windows type in "arp -a" at the command line) to see if it already has the MAC address, but if not it broadcasts out an ARP request asking "Yo, who has this IP address I'm looking for?" If the host that has that IP address hears the ARP query it will respond with it's own MAC address and a conversation can begin using IP. In common bus networks like Ethernet using a hub or 801.11b all traffic can be seen by all hosts who's NICs are in promiscuous mode, but things are a bit different on switched networks. A switch looks at the data sent to it and tries to only forwards packets to its intended recipient based on MAC address. Switched networks are more secure and help speed up the network by only sending packets where they need to go. There are ways around switches though ;). Using a program like Arpspoof, Ettercap or Cain we

can lie to other machines on the local area network and tell them we have the IP they are looking for, thus funneling their traffic through us.



The image on the left helps illustrate how arpspoofing/arppoisoning works. Basically, the Cracker is telling Alan's box that he has the IP that corresponds to Brian's box and vice versa. By doing this the Cracker receives all network traffic going between Alan and Brian. Once you have Arpspoofed your way between two machines you can sniff the connection with whatever tool you like (TCPDump, Ethereal, Ngrep, etc.) By arpspoofing between a machine and the LANs gateway you can see all the traffic it's sending out to the Internet. In this tutorial I'm only giving the basics of how to use these tools, look at their specific MAN pages and documentation for a plethora of more advanced options.

Dsniff Tools (arpspoof)

Lets start with using Dug Song's Arpspoof program that comes with his Dsniff (<http://www.monkey.org/~dugsong/dsniff/>) package. I'll be using the *nix version but if you look around you may be able to find a Win32 version. First thing we should do is make sure packet forwarding is turned on, other wise our

machine will drop all traffic between the hosts we are trying to sniff, causing a denial of service. Some of the tools I'll show do this for you automatically, but to be sure you may want to do it yourself. Use the following commands, depending on operating system:

Linux:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

BSD:

```
sysctl -w net.inet.ip.forwarding=1
```

Now that our box will forward the traffic we can start Arpspoofing. Let's assume I want to sniff all traffic between a host and the gateway so I can see the traffic it's sending to the Internet. To get traffic in both directions I would use the following two commands:

```
arpspoof -t 192.168.1.1 192.168.1.2 & >/dev/null
```

```
arpspoof -t 192.168.1.2 192.168.1.1 & >/dev/null
```

The "& >/dev/nul" part is there to make it easier to run from one terminal but you may want to omit it for debugging purposes. Now we can use any package we wish to sniff the connection. To start with I'd recommend using the sniffer dsniff that comes along with arpspoof to sniff for plain text passwords. To look at all sorts of other traffic I would recommend TCPDump or Ethereal. When you are ready to stop arpspoofing issue the following command.

```
killall arpspoof
```

This should kill the two instances of arpspoof started above.

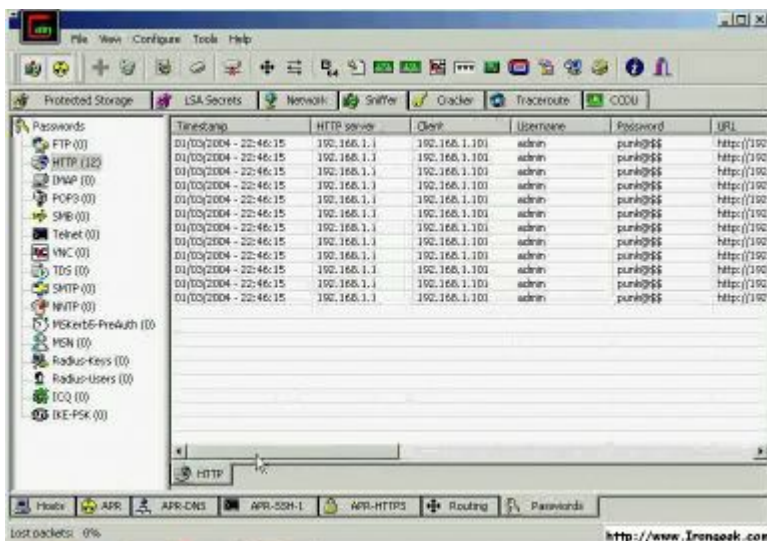
Ettercap

Another package you may want to look into is Ettercap (<http://ettercap.sourceforge.net/>). It's sort of the Swiss army knife of Arpspoofing and password sniffing. I usually use it in non-interactive mode, but by default it has a ncurses interface. Here's a quick example of how to sniff for passwords in non-interactive mode between two machines.

```
ettercap -NaC 192.168.1.1 192.168.1.2
```

The "N" option makes it non-interactive, the "a" option tells it to arppoison and the "C" tells it to parse out passwords and usernames. Ettercap and Dsniff are both great tools for sniffing passwords on protocols that send them plaintext (telnet, SMTP, http, etc.) A nice thing about Ettercap is that it will proxy some connections like SSL and allow you to sniff traffic that is usually encrypted, the victim will get a warning message about the certificate, but many folks just click past such things without reading them. If you want to use Ettercap to just arpspoof so you can use another sniffing tool just look in the man page for the "-J" option.

Cain



For you

Windows users, look into using Cain (<http://www.oxid.it/cain.html>). It has some great functionality. I have a video tutorial on how to use it here: <http://www.irongeek.com/i.php?page=videos/cain1>

If you like pretty GUIs, Cain is the way to go. It does not have as many options as Ettercap, but it's still pretty cool and has some other Windows specific extras built in.

Further research:

Protecting yourself from arpspoofing

Static ARP tables

ARPWatch (<http://www-nrg.ee.lbl.gov/>)

Also see my article on how to [change your MAC address](#)