

Notes on running Dsniff in the lab

Cyber Security Lab
2/21/2006

Arpspoof

You can use arpspoof to poison a target's ARP cache, so it will use the attacker's MAC address. This will route all traffic from the victim machine through the attacking machine, enabling sniffing which is not normally available in a switched environment.

The following command

```
arpspoof -t 192.168.50.16 192.168.50.1
```

will poison the arp cache of the machine with the address 192.168.50.16, so it will think the attacker machine is the default gateway (192.168.50.1). The arp cache clears pretty quickly (in a couple seconds), so you must keep the arpspoof program running during your attack.

The attacker machine must turn on IP Forwarding. This can be controlled through the proc pseudo file system. Change the contents of the file `/proc/sys/net/ipv4/ip_forward` from 0 to 1.

You do not need to explicitly poison the gateway in our situation. It uses the MAC address from the initiating packet (which will be the attacker's MAC) when filling in its session table. So the reverse traffic will pull the attacker's MAC from the session entry. In other situations, you would need to run arpspoof a second time to poison the gateway's arp cache.

Now traffic from 192.168.50.16 should pass to the outside world and return traffic should come back. Both the forward and the return traffic will pass through the attacking machine.

Dnsspoof

Arpspoof must be running to ensure that the DNS request passes through the attacking machine. Or the machine must be sitting on a span port. Dnsspoof will intercept all DNS requests and reply with the attacking address. Note that you need a 2.4 version of dsniff. It looks like dsniff 2.3 has a byte order problem. The windows machine would see the dns spoofed address backwards.

You can see the results of the dnsspoof by doing a ping from the victim machine, e.g. ping to Microsoft.com should show that it is really pinging the attacking machine.

Webmitm

At this point it is obvious how you could write man in the middle (MITM) tools. The Dsniff package includes several MITM tools. I played with webmitm with mixed results. Webmitm helps you create a certificate. It then intercepts web traffic and passes it onto the ultimate destination. For non-SSL traffic it will pass traffic and print out passwords. For SSL traffic it will present its certificate. The browser will print many warning messages. If you click through them it will show you the first SSL page. But after the first page, the SSL interactions seem to stall out. I guess that the SSL version in webmitm (built in 2001 or 2002) does not match the versions that are currently being used.