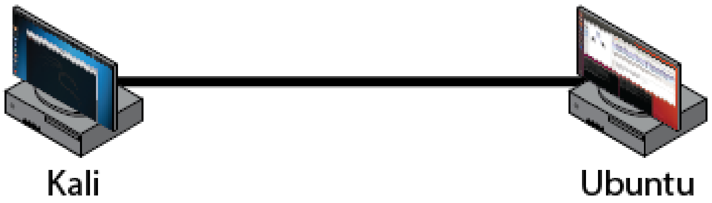
SIT716 Computer Networks and Security

Week 7: Tunnelling

Below you will find instructions for conducting an experiment designed to explore concepts of networking and/or security. Before referring to the instructions below, make sure you take the time to research and design your own experiment – even if you can't work out everything that is required. After completing your study of this unit, the skills you develop in designing your own experiments will be a critical skill for learning about new network protocols and security attacks, enabling you to work successfully as a valued professional in these fields.



Host	IP Address	Role
Ubuntu	192.168.1.1/24	Sender
Kali	192.168.1.2/24	Receiver

Outline:

In this lab we will explore the IP-in-IP and GRE tunnelling protocols and SSH tunnelling. IP-in-IP and GRE tunnels provide no encryption, whereas SSH provides an encrypted tunnel for a single application protocol to be transported to another host.

Instructions:

1. Load VirtualBox, Ubuntu, and Kali.
2. Ensure both Ubuntu are connected to the internet network. Refer to the 'Connect Kali and Ubuntu' screencast in Week 4 if you aren't sure how.

Prepare Ubuntu

We begin by preparing Ubuntu as one end of the tunnel and start the DVWA application to provide a target.

1. On Ubuntu, begin an instance of DVWA. Enter in the following command in a new terminal: (skip this step if you are using the VMs provided on-campus, it's already running)

   ```
   sudo docker run –rm -it -p 80:80 vulnerables/web-dvwa
   ```

2. Prepare the Ubuntu end of the IP-in-IP tunnel: (note that you can also complete this experiment replacing 'ipip' with 'gre' for a GRE tunnel)

   ```
   sudo modprobe ipip tun
   sudo ip tunnel add tun0 mode ipip remote 192.168.1.2 local 192.168.1.1
   sudo ip address replace 10.0.0.1/24 dev tun0
   sudo link set tun0 up
   ```

Prepare Kali

1. Prepare the Kali end of the IP-in-IP tunnel: (note that you also complete this experiment replacing 'ipip' with 'gre' for a GRE tunnel)

   ```
   sudo modprobe ipip tun
   sudo ip tunnel add tun0 mode ipip remote 192.168.1.1 local 192.168.1.2
   sudo ip address replace 10.0.0.2/24 dev tun0
   sudo link set tun0 up
   ```

2. Create the SSH tunnel to Ubuntu for remote port 80 (local port 2000):

   ```
   sudo ssh -f your_ubuntu_username@192.168.1.1 -L 2000:192.168.1.1:80 -N
   ```

Examine the Traffic Resulting from the Tunnels

1. Start an instance of Wireshark on Ubuntu and begin tracing network traffic:

   ```
   sudo wireshark
   ```

2. In Kali, start a web browser and connect to the Ubuntu web server using three different addresses (note that you don't need to setup DVWA for this activity):

   (a) http://192.168.1.1
   (b) http://10.0.0.1
   (c) http://localhost:2000

3. Return to Ubuntu and analyze the traffic captured in Wireshark. You should be able to see three sets of traffic according to the addresses used (a) direct access (as per previous weeks), (b) through the IP-in-IP tunnel (or GRE tunnel if you have used that), and (c) through the SSH tunnel. Examine the differences between these three sets of traffic.

Note: if you aren't able to see the difference between the traffic types clearly, try restarting the packet capture in Wireshark, then access the web site using only one of the URLs. Review the Wireshark capture, then restart the capture and repeat using a different URL. This way you isolate the different types of traffic.