Helsinki University of Technology
Department of Electrical and Communications Engineering
Networking Laboratory

S-38.3133 - Networking Technology, laboratory course

Spring 2007
Work number 37: Wireless Networks

Instructions, questions for preliminary and final reports.

16.8.2005 and revised 3.1.2007
Anni Matinlauri

Work 37 – Wireless Networks

Preliminary exercises

*Answer the following questions shortly but **clearly**. You can answer in Finnish/Swedish or in English. It is also a good idea to examine the laboratory assignment beforehand. There is only 3 hours work time on your lab turn.*

P1. Is it legal to listen to unencrypted WLAN traffic?

P2. Is it legal to collect encrypted WLAN data and crack it?

P3. What are the ways to make WLAN safer?

P4. Describe shortly what is an Ad hoc network?

P5. Explain briefly how mobile IPv6 works?

P6. What kinds of wireless standards does IEEE have?

P7. What security improvements have taken place as IEEE 802.11 standards have developed?

P8. How does WEP work?

Background info

As wireless networks are gaining popularity it is important to get to know them. In this assignment the main focus is in security aspects. Some hacking techniques are tried so that you understand in which ways wireless networks are vulnerable. These assignments however, by no means cover everything and every new way to exploit wireless networks, so it is important to always follow the latest development.

The most important commands in this assignment are ifconfig and iwconfig. If you are not familiar with them check their man pages.

The strong present of Aalto network it may disturb your connection to Matti and Teppo access points. In case some of the machines suddenly try to attach to Aalto. Just make them go back to netlabtest by writing iwconfig ethX essid netlabtest.

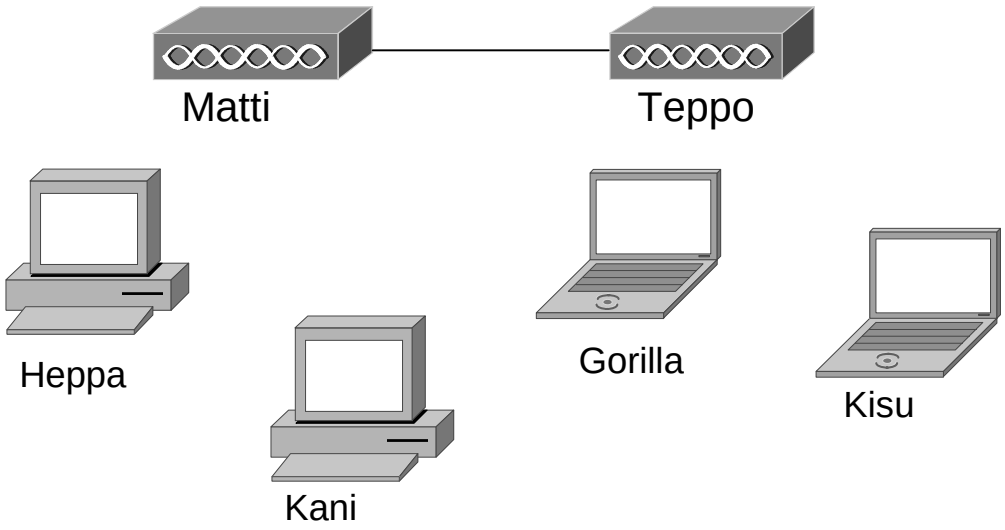
Also don't worry if everything does not work when you first try it. Making the machines connect to APs may take some effort. You might need to remove the WLAN cards on laptops and try again and rewrite commands several times.

In your final report answer questions Q1-Q13 and final questions F1-F3. Keep your answers short but precise.

Assignments

Scanning WLANs

The topology of the network is shown in the figure below:



Log on to Kisu as labra. Set the IP address of the wlan interface to be 10.38.40.101/24. Log on to Gorilla as labra. Set the IP address to be 10.38.40.100/24. Have both machines connect to the network netlabtest. Start pinging from Gorilla to Kisu.

Log on to Kani as labra. Put the wlan card to monitor mode. Use kismet on Kani to scan for possible wireless networks.

Q1: Write down the following information about networks other than aalto: SSID, BSSID, channel, IP range and signal power.

Log on to Heppa as labra. Put the wlan card to monitor mode. Use airsnort to scan.

Q2: What information does the program give? Compare shortly Kismet and Airsnort.

End the ping between Gorilla and Kisu. Transfer with scp the file named aputiedosto in /home/labra direcotory between Gorilla and Kisu.

F1a: Write down the transfer rate and the amount of time it took to transfer.

Posing as an access point

Use the information you just collected to pose as an AP. Change Heppa to be in AP mode and change its channel and SSID to be the same as a netlabtest AP's. Write down Heppa's old mac address. Then change it to be the mac of a netlabtes AP using ifconfig. Change Heppa's IP address to be 10.38.40.103/24 and ping this address from Gorilla and Kisu to make sure they are connected to Heppa.

Q3: What harm could you do in real life if you managed to pose as an AP?

Fake Access Points

To fool possible attackers it is possible to generate false AP signals. In Heppa there is a program fakeap in the /home/labra/fakeap directory. Use this program to generate fake access points. Have some of the fake AP's use WEP. Also for ESSIDs use the wordlist in /usr/share/dict/words directory and for mac addresses use the list in /usr/share/ethereal/manuf directory. Change also the time the AP's exist to be longer than default. Use Kismet on Kani to monitor the situation.

Q4: What does Kismet show?

End the fakeap program and have the laptops connect to the access points again.

WLAN power

Write vlc --ipv4 http://stream.noc.lab:7000 (IP address 130.233.154.61) on Gorilla. Music streaming should now work. Connect to the access points Matti and Teppo as root (10.38.40.2 and 10.38.40.1) use iwconfig or wl txpwr to modify their power levels. Find out what is the minimum power where each access point is still audible?

OR

Alternatively if the streaming does not work, put Matti's signal power to zero. Then start to ping from one of the laptops to Teppo. Pick up the laptop and see how far you can go with the ping still going through.

Q5: What can you say about the results?

Q6: Why is it good to modify an AP's power level?

Finally put the power levels of AP's back to normal.

WEP

Connect to AP Matti (10.38.40.2) as root. The APs are using OpenWrt OS which is a barebones Linux, a fast Linux based firmware for the Linksys WRTs.

Enable WEP and use 40 bit encryption. Set the key to be ABCDEFABCD. This can be done by writing wl wep on and wl addwep 0 ABCDEFABCD. In order for the changes to happen write wlconf eth1 down and wlconf eth1 up. Also Connect to Teppo (10.38.40.1) and write wlconf eth1 down. This way Teppo's signal won't disturb this exercise. Check with the command iwconfig that everything WEP is on.

Start generating traffic with flooding ping or file transfer. There is a file named bigfile in /home/anni/p2p/JXTADemo directory. Use the machine Kani (or Heppa) to collect traffic. In Heppa and Kani there is a program Airodump which can be used to collect traffic. For 64 bits collect over 300 000 IVs. Then use the program Aircrack to crack the WEP. If you're out of luck you need to capture more packets. Newly captured files can be merged with old files by using the mergecap program. If it seems that the collecting is taking far too long, then make an estimate of the time you think it would take.

Q7: How long did the data collecting take or did seem to take?

Q8: How long did the actual cracking take? (If you were unable to capture enough packets, use the previously captured file sieppaus64)

Similar process can be used when cracking WEP with longer keys, only the amount of IVs needed is higher. For 128 bit WEP over 1000000 IVs are typically needed. Use the previously captured file sieppaus128 to crack 128 bit WEP.

Q9: How long it takes for the program Aircrack to crack 128 bit WEP?

Q10: What can you say about the security of WEP?

Q11: Use the file for 64 bit crack and open it with Ethereal. What can you see? Then decrypt the file with 802ether and use Ethereal again. What can you see now?

F1b: While 64 bit WEP is on, measure the transfer rate and the amount of time it takes to transfer the file aputiedosto from Gorilla to Kisu.

Ad hoc network

Now switch all the machines to ad hoc mode. Set Kani's IP to be 10.38.40.102/24 and Heppa's to be 10.38.40.103/24.

F1c: Again transfer the file aputiedosto between Gorilla Kisu and write down the transfer rate and transfer time.

Next you will test two Ad hoc protocols, AODV and OLSR. The code for these protocols is experimental and hence not very user friendly. Problems may occur but try to be patient and methodical. Ask the assistants help if necessary. The most important thing

here is to try both protocols at least directly. Getting the transfer through hops can be trickier, so if you cannot do it that is okay. However, extra points are available for one hop and two-hop transfer results.

First put down the eth0 interface of Heppa, it can interfere. Then start Ethereal in one of the machines and start to capture. Finally start OLSR routing with the command olsrd.

Q12: Analyze your OLSR capture.

Now transfer aputiedosto from Gorilla to Kisu.

F1d: What are the transfer rate and time of aputiedosto now?

Next, use iptables firewalls to force the transfer to use more hops. The forcing can be done by blocking mac addresses. Important commands:
iptables –L
iptables –A INPUT –m mac –mac-source <mac address> -j DROP
iptables –F

First measure a transfer with one hop.

F1e: What are the transfer rate and time of aputiedosto now?

Now have the transfer go through two hops.

F1f: What are the transfer rate and time of aputiedosto now?

Now end OLSR routing and start AODV routing in /home/labra/aodv-uu-0.9.1 directory with the command aodvd and correct switches. Use Ethereal again to capture what happens as the routing starts.

Q13: Analyze your AODV capture.

Transfer aputiedosto directly from Gorilla to Kisu.

F1g: What are the transfer rate and time of aputiedosto now?

Again force the transfer to go through one hop.

F1h: What are the transfer rate and time of aputiedosto now?

Now force the transfer to go trough two hops.

F1i: What are the transfer rate and time of aputiedosto now?

Final Questions

F1: Make a graph depicting all the transfer rates and times you have collected marked with F1a-F1i. What can you say about the results? How quickly does the connection deteriorate if there is more than one hop?

F2: During the lab you saw how WEP can be cracked. What about WPA? Discuss briefly WPA's security.

F3: Ad hoc routing protocols can be divided into reactive and proactive protocols. Explain the difference and specifically explain how AODV and OLSR work?